NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Cased Based Reasoning in Business Process Management Design

Philipp Tueschen

Dissertation presented as partial requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

CASE BASED REASONING IN BUSINESS PROCESS MANAGEMENT DESIGN

By

Philipp Tueschen

Dissertation presented as partial requirement for obtaining the Master's degree in Information Management

Advisor: Professor Vítor Manuel Pereira Duarte dos Santos

May 2021

ACKNOWLEDGEMENTS

I would like to use this opportunity to acknowledge the people that have made this graduation project possible and have supported me along the way.

I would like to thank Professor Vítor Manuel Pereira Duarte dos Santos for the valuable guidance throughout the entire duration of the thesis. Without his encouragement and knowledge sharing the dissertation would not have been possible. I would also like to thank the interviewees Isabel Machado Alexandre, Pedro Maia Malta, and Frederico Cruz Jesus for their time and constructive feedback on the model.

Lastly, I acknowledge with immense gratitude my family, who supported me continuously. They have always given me the freedom and encouragement to do my best. Thank you for helping me grow into the person I am today.

ABSTRACT

Artificial intelligence became increasingly useful since the 1990s, trying to imitate the human brain with its thinking, reasoning, and learning using the key concepts of machine learning, deep learning, and artificial neural networks. Case-based reasoning (CBR), another form of artificial intelligence, stores and retrieves past cases that can be adapted to find a solution to a current problem. The new solution can then be retained and made available to solve other future problems. Business Process Management (BPM) analyzes and optimizes business processes to make them more effective and efficient for an organization's strategy to ultimately increasing shareholder value. CBR can help to support BPM, making better decisions with existing knowledge when solving process problems. This study investigates effectively store, retrieve, and adapt Business Process Management Notation (BPMN) solutions that best fit the underlying BPM problem using case-based reasoning as a tool. Therefore, a theoretical model was proposed, containing each CBR live cycle phase with different possible tools applied to BPMN diagrams, which was validated by expert interviews. This study concludes that a whole CBR life cycle can be applied to BPMN diagram problems with the need for human intervention. This work did not have the objective to solve the whole problem but to contribute to a possible solution by using CBR through a theoretical model.

KEYWORDS

Artificial Intelligence; Business Process Management Notation; Case-Based Reasoning; Design Science Research; Graph-Edit Distance; Learning by Analogy; XPDL

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
BPM	Business Process Management
BPMN	Business Process Management Notation
BPMS	Business Process Management System
CBR	Case Based Reasoning
DSR	Design Science Research
GED	Graph Edit Distance
GES	Graph Edit Similarity
MEA	Means-End Analysis
RBES	Rule-Based Expert System
SCEM	Supply Chain Event Management
XPDL	XML Process Design Language

PUBLICATIONS RESULTING FROM THIS DISSERTATION

The Master Thesis “Cased Based Reasoning in Business Process Management Design” was accepted as a paper for Artificial Intelligence in Intelligent Systems – Proceedings of 10th Computer Science On-line Conference 2021.

1 INTRODUCTION

Since the 90s artificial intelligence (AI) is becoming more and more useful; automating more and more tasks makes human life easy by copying the human way of learning, thinking, and reasoning. AI can conduct complicated tasks, such as autonomous driving. This imitating of the human brain is accomplished based on the three basic concepts within the AI area: machine learning, deep learning, and neural networks (Sciglar, 2018).

Case-based reasoning (CBR) is part of the AI area as CBR learns by storing initial knowledge and retrieving it when needed. Further, CBR reuses this knowledge as a possible solution, which is then revised to solve a current problem. In the end, a successful solution to a problem is retained in the case database adding new knowledge to the CBR system, which is considered "learning" (Pantic, 2006).

Business process management (BPM) problems rely on expert's knowledge to resolve them. As BPM focuses on finding improvement opportunities and managing processes, consistency experts rely on their past experiences to tackle the present problems.

CBR is imitating the expert's knowledge acquisition by storing knowledge from the past and using applicable knowledge to solve a present problem, making it a good system to manage BPM problems more efficiently. In fact, Pichler (2011) found that using CBR implemented in a BPM system allows them to execute processes that have clearly defined and standardized process models and react to events outside the limitations. Pichler (2011) used Petri nets as a similarity function within the CBR system to find the most suitable cases to the underlying problem. However, this report will work with Carbonell's (1985) learning by analogy, which compares semantics and considers the decisions' justifications of positive and negative outcomes. Further, Carbonell's "learning by analogy" is incorporated into the CBR system, supporting the retrieval and adaptation process.

1.1 MOTIVATION

Although Pichler (2011) establishes that CBR can help to find solutions for BPM problems by using a similarity function establishing the distance between the attributes of the business process management notation (BPMN) diagrams, the article does not talk about the storing and retrieving of BPMN diagrams, which is an essential part of CBR in providing the best fit solution.

Existing literature established the possibility of translating BPMN models into XPDL language and that CBR can be an effective method for improving BPMN solutions (White, 2003; Pichler, 2011). However, there is a gap between how CBR could be used in comparing BPMN models and how CBR used with BPMN could be useful with different models then proposed by Pichler (2011). Hence, this thesis aims to find a solution to how BPMN can be stored effectively in

XDPL and how the different stored XPDL files can be analyzed and retrieved effectively. This thesis also considers the BPMN adaptation, providing the best fit solution the user is looking for. This thesis's outcome is abstract knowledge; however, it could contribute to a more physical solution in the future (Gregor & Hevner, 2013).

1.2 OBJECTIVE

Having identified the gap between CBR usage combined with BPMN, this paper aims to **effectively store, retrieve, and adapt BPMN solutions that best fit the underlying BPM problem using case-based reasoning as a tool**.

To understand the problem and develop a sound solution, understanding how the two main topics, namely, BPMN and CBR, can work together effectively is essential to get the desired outcome of delivering correct BPMN solutions. Therefore, sub research questions, as intermediate steps, guide deriving an answer on how to store, retrieve and adapt the right BPMN solutions:

1. How can a BPMN diagram be efficiently stored?
2. How many attributes describing the diagram should be stored?
3. What technology can be used to compare BPMN diagrams?
4. How can the technology identify the correct semantic of different words having the same meaning?
5. How can the best match be determined?
6. What is the best way to adapt the best match to the current problem?

1.3 THESIS ORGANIZATION

The thesis is structured into six parts. The proposal needs three parts due to the utilization of the CBR life cycle. The unique parts of storing, retrieving, and adaptation need different and unique approaches to bring the whole CBR life cycle together as one working solution. Additionally, as retaining new solutions is the function of storing, it is integrated into the adaptation section. For that reason, the six parts are following:

1. Definition of Goals
2. Literature Review
3. Methodology
4. Proposal Storing
5. Proposal Retrieving
6. Proposal Adaptation

2 LITERATURE REVIEW

2.1 ARTIFICIAL INTELLIGENCE

As the word already implies, AI aims at imitating human intelligence, such as learning, reasoning, and alteration.

"AI forms a theoretical and methodological basis for learning symbolic representations of concepts, learning in terms of classification and pattern recognition problems, and learning by using prior knowledge together with training data as a guideline (Pantic, 2006, p.3). "

Since the 90s, AI's broad area has realized much progress in fields such as big data and self-driving vehicles. This progress is based on the three basic concepts of machine learning, deep learning, and neural networks. Machine learning focuses on enabling machines to learn independently, based on a fast number of trial examples through which the machines learn to adapt. Deep learning concentrates on using the learned to apply it in a different area and consequently requires general-purpose learning algorithms to perform more than one task. This learning process is enabled by artificial neural networks (ANN) that use biology to mimic brain cells constructed by code. Hence, these three concepts permit software to "think" and react flexibly using algorithms that calculate and analyze the best outcome (Sciglar, 2018).

Another area in AI fitting the definition by Pantic (2006) above is called an expert system. This system directly focuses on predefined and described knowledge retrieved if it fits to solve a specific problem. Once the problem is solved, the solution is stored for the future, which is comparable to "learning". These expert systems developed further due to any identified problems as the knowledge is now stored and used for a specific problem, called case-based reasoning (CBR) (Corchado, Lees, 2001).

2.2 LEARNING BY ANALOGY

According to Carbonell (1985), learning by analogy and solving problems initiates with the reminding process using the standard Means-End Analysis (MEA), which consists of four steps:

1. Current state to goal state comparison
2. Operator choice to reduce the difference
3. Application of the operator if possible
    a. If the application is not possible use MEA to solve the subproblem of an unsatisfied precondition of the operator
4. Return to the current state and continue with the original problem

For the difference function in MEA, Carbonell suggests a similarity metric to find already solved problems closely related to the current problem. Each solution is built by sequences of operators, initial and final states. The difference function has to take into account the current problem's path constraints and different operator sequences. In a second step, the old solution must be adapted to the new problems satisfying its criteria (Carbonell, 1985).

2.3 CASE-BASED REASONING

In the 80s, AI research focused on rule-based expert systems (RBES) that use their induced knowledge to reason using a specified set of principles. However, these RBESs had several problems regarding the requirement for comprehensive knowledge. First, the RBESs are very time-consuming in construction because of the need for expert knowledge. Further, if the system's knowledge does not cover the problem, it cannot be handled. Lastly, if the system cannot learn by itself, any addition to the current knowledge requires a programmer (Pantic, 2006).

CBR picks up the expert system's problems as it uses and adapts solutions that have been successful in the past to solve new problems. Case histories are collected for CBR, and their most important features are described. New solutions of cases are added as new knowledge to the CBR databases. Thus, as databases enable the management of larger volumes of cases, it makes the need for explicit models obsolete (Pantic, 2006). All in all, CBR uses successful past cases to suggest a solution to a current, new to the system, and similar problem. Kolodner (1996, as cited in Pantic, 2006) came up with four assumptions based on CBR:

1. *Regularity:* actions performed under identical condition will have comparable outcomes
2. *Typicality:* Experience repeats itself
3. *Consistency:* Small differences in the actions performed only need small changes in the solution
4. *Adaptability:* Repeatable things gravitate towards small differences

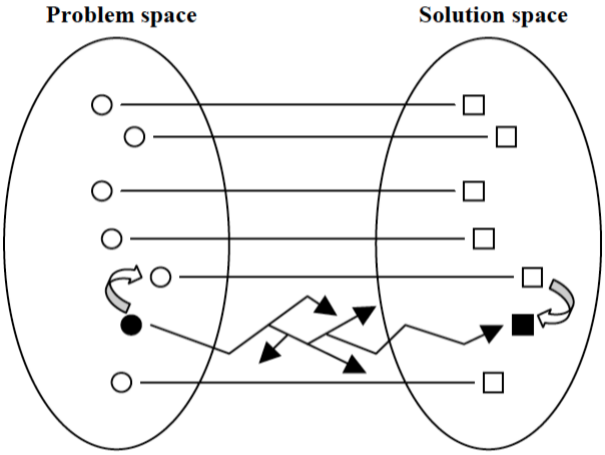

Figure 1 - CBR Problem Solving (Pantic, 2006)

These assumptions used by CBR are illustrated in Figure 1. After the current problem case could be described with its features, the most similar case stored can be allocated. Although this stored solution may not directly fit the current problem case's differences, it can be adjusted. The derived and authenticated new solution to the current problem can then be stored as a new case solution to solve a similar problem in the future (Pantic, 2006).

2.3.1 Case-Based Reasoning Working Cycle

The process shown in Figure 1 builds the ground for the CBR working Cycle. The CBR working cycle (see Figure 2), according to Aamodt and Plaza (1994), is defined as the four Res:

1. *Retrieve:* After a current problem has been defined the most applicable solution is searched for and obtained from the case database
2. *Reuse:* The solutions obtained and used to solve a current problem
3. Revise: The obtained solution has to be adapted to the current problem to build a possible solution. If the solution is undesirable further adopting is required.
4. Retain: If the solution is desirable and verified as such it can be added as a new case to the database

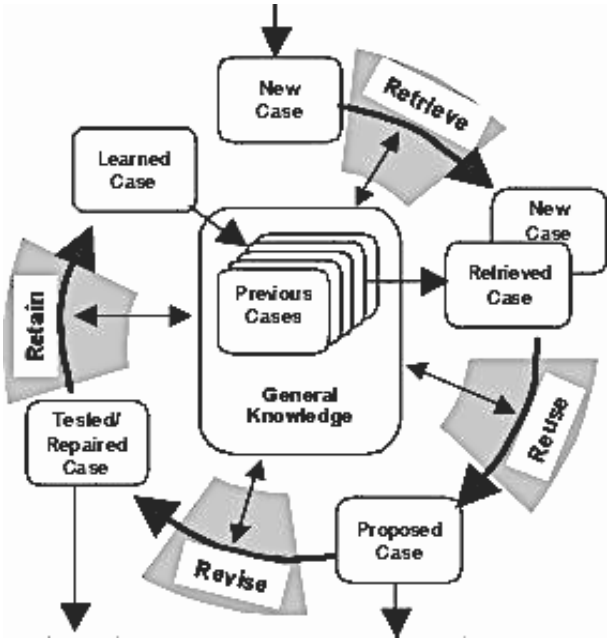

Figure 2 - The CBR Cycle (Kolodner, 1995)

2.3.2 Derivational Analogy

Solving a problem with derivational analogy means to search for related problems in the database and adapt their solutions to be possibly relevant to the current problem, which is why Carbonell (1985, p.3) defines it as the following: *"Analogical problem solving consist of transferring knowledge from the past problem-solving episodes to new problems that share significant aspects with the corresponding past experience - and using the transferred knowledge to construct solutions to the new problems."*

Carbonell points out that it is not enough to find a solution similar to the current problem but rather about the knowledge, it contains, which can be retrieved and interpreted. It implies the necessity for a very detailed adaptation model to obtain a desired adapted solution from the past. As this way of problem-solving is a fundamental part of human cognition, the derivational analogy is undoubtedly part of AI (Carbonell, 1985).

Building a model to adapt past case problems to the current situation requires specificity. The model needs to know what it means for both cases to have meaningful aspects in common. Further, what knowledge from these aspects is transmitted from the past solution to the current problem and how it occurs. Finally, it is important to know how these analogically associated aspects are selected from the case base (Carbonell, 1985).

The last part can be translated into the following outline that should help to analyze and compare current and past situations in a structured manner. Past solutions and current problems have meaningful common aspects if their initial stages or segments (the beginning part of the problem's solution) share the same issues and decisions. Therefore, additional information, such as the justification of the decisions taken, must be considered. Once the two problems commonly justify the same issues and decisions, they can be transferred from the old to the current problem (see Figure 3). It is important to mention that the derivation's reasoning process that solved the last problem is recreated in the current situation. Hence, the knowledge transfer enables reassessing old decisions and their resulting reasoning process if they are still suitable for the current situation (Carbonell, 1985).

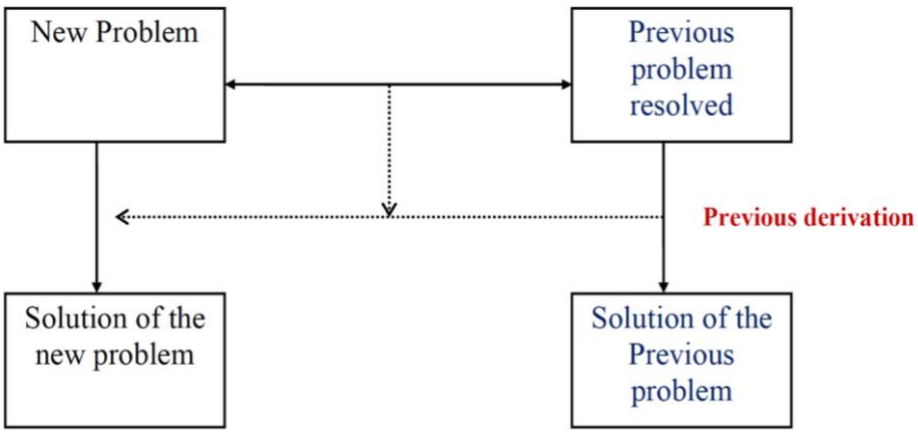

Figure 3 - Derivational Analogy (Carbonell, 1985)

Building general plans help to automate the acquisition of the past problem's transferrable segments. Therefore, the past solutions derived from the analogically related problems to the current problem need to have positive and negative examples. The examples should justify a successful decision or show inefficiencies where examples had a negative outcome. The cause of the negative decision's ineffectiveness can distinguish between good and bad instances of

the initial stages' decisions. These solutions can then be fed to a general inductive engine that abstracts a general plan from all the solutions' common aspects (Carbonell, 1985).

After the general inductive engine abstracted a general plan, it can now find relevant segments of past problems. Further, it is now possible to determine their decision's justifications with matching preconditions. These matching justifications are then utilized to formulate rules that help learn more general knowledge applicable to current situations. However, the difficulty in derivational analogy consists of the precondition of all derivations' availability to form a solution (Kuchibatla & Muñoz-Avila, 2006).

2.3.3 Transformational Analogy

Carbonell (1983, as cited in Kuchibatla & Muñoz-Avila, 2006) depicts transformational analogy as taking part in the CBR adaptation where a solution and its sequence of actions from a previous problem is being modified into the solution of the new problem (see Figure 4). The modification includes removing, adding, and/or changing the sequence's actions. Hence, after the closet solution has been found, the derivational and transformational analogy can be used as a CBR strategy to form a reasoning process that helps adapt the old solution to the current problem (Kuchibatla & Muñoz-Avila, 2006).

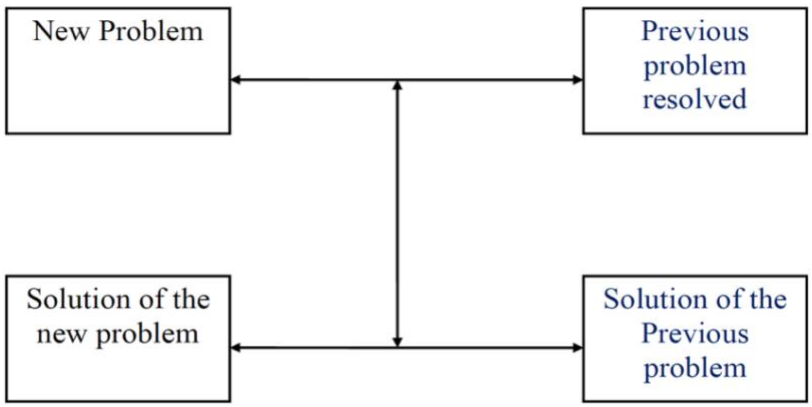

Figure 4 - Transformational Analogy (Carbonell, 1985)

2.4 Business Process Management

BPM analyses operational activities performed in organizations and identifies advantages of improving processes, guaranteeing their consistency. Thus, BPM helps operating processes faster, more accurately, at lower costs, and with reduced assets while increasing their flexibility. Focusing on End-to-End processes across organizational boundaries can reduce nonvalue-adding tasks, which exist due to departmental boundaries. Further, processes are continuously monitored and improved if they no longer meet the customer's needs (Hammer, 2015). There are several methodologies with tools and techniques that help to identify redundant, inefficient, and ineffective processes or highlight specific processes for

improvement, such as Six Sigma, which is using statistical models to analyze processes without directly assessing end-to-end processes (Van Der Aalst, La Rosa, Santoro, 2016).

The BPM as a management discipline can be divided into two approaches. Process improvement aims to analyze existing processes and consciously improve them while process reengineering questions existing processes from end-to-end and redesigns them from scratch. Therefore, BPM does not create business value by merely using IT or information systems to automate processes, but BPM uses IT and information systems to enable process change, creating business value (Dumas, La Rosa, Mendling, Reijers, 2013; Hammer, 2015).

"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency. "

— *Bill Gates*

2.4.1 Process Structure

A business process is a set of structured tasks creating a service or a product that satisfies a specific need of one or many actors. This process is composed of events that happen automatically with no time duration and activities that need some action involving time duration.

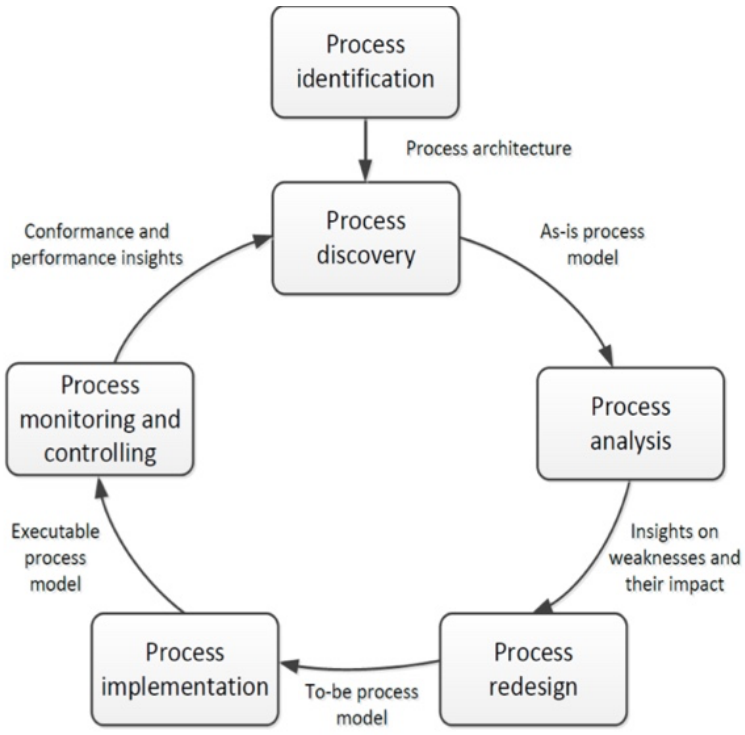

Figure 5 - BPM Life Cycle (Dumas et al., 2013)

Further, decision points affect the outcome of the process and actors, such as humans or physical objects. The outcome can be either positive (desirable) or negative (undesirable) depending on if it creates value to the actors involved. The outcome of a process is used by an internal or external customer of the organization. Based on these key concepts, Dumas et al. (2013, p6) defines a business process as *"a collection of inter-related events, activities and decision points that involve a number of actors and objects, and that collectively lead on an outcome that is of value to at least one customer"*. Identifying business processes, their discovery, analysis, redesign, implementation, controlling and monitoring, is called the BPM Life Cycle, as shown in Figure 5 (Dumas et al., 2013).

2.4.2 Core Elements of BPM

BPM needs a framework structuring and decomposing BPM after understanding the basic ideas of how to design, manage, and change processes for different organizational reasons. Therefore, Rosemann and vom Brocke (2010) outline six core elements for BPM that have to be addressed for sustainable and favorable progress. Strategic alignment links organizational priorities and enterprise processes; governance transparentizes roles and responsibilities on different BPM levels. Methods provide the tools enabling ongoing operations on all levels, while IT provides the hardware and software. People's knowledge and skills continually enhance performance and culture that summarize the values and beliefs shaping process-related tasks and the environment. Keeping these six core elements in mind is a critical factor for a successful BPM.

2.4.3 Different Business Process Modeling Languages

There are many different modeling languages nowadays for BPM and its visualization. Therefore, the five most recognizable languages have been chosen according to Pereira and Silva (2016). These are the Business Process Model and Notation 2.0 (BPMN 2.0), which is considered the standard, the Event-driven Process Chain (EPC), the Unified Modeling Language – Activity Diagrams (UML-AD) created by the Object Management Group, the Integration DEFintion (IDEF), Role Activity Diagram (RAD). Pereira and Silva (2016) then evaluated the languages and compared them with each other. The outcome can be seen in Figure 6, where zero means no support and five means full support of the criteria. It becomes clear that IDEF has the most limitations, while BPMN has the highest acceptance, supporting the fact that BPMN is the most accepted and widely used language for BPM (Pereira & Silva, 2016; Jung, Kim, Jo, Tak, Cha, & Son, 2004).

After identifying BPMN 2.0 as the most popular language for BPM, it can be analyzed in more detail. Although BPMN 2.0's evaluation depends on the modeling task, to evaluate BPMN 2.0's quality, the following six categories are used, called SEQUEL, developed by Krogstie, 2012a (as cited in Aagesen & Krogstie, 2015).

Languages / Criteria	BPMN	EPC	UML-AD	RAD	IDEF
Expressiveness	4	3	4	3	2
Readability	5	4	4	4	3
Usability	4	4	4	4	3
User Friendly	5	5	5	5	3
Formality	5	5	5	1	5
Versatility	5	5	4	3	3
Universality	5	4	5	3	3
Tools Support	5	2	5	2	3
Flexibility	4	4	4	4	3
Concision	4	4	4	4	3
Ease of Learning	5	5	5	4	3
Innovation Inducer	4	4	3	5	2
Evolutionary	4	4	4	2	3
Collaborative Work	2	2	2	5	0

Figure 6 - BPM Language Evaluation (Pereira & Silva, 2016)

Regarding the domain appropriateness, the language has limitations in resource modeling and includes business rules and data, which is only supported on a high level. The comprehensibility appropriateness shows multiple redundant representations for the same patterns while lines and pools can be cluttered. When it comes to the modeler appropriateness Aagesen and Krogstie (2015) point out the excessive use of text annotations in imprecise models to substitute their expressive power instead of using more meaningful language constructs. Under participant appropriateness, it is mentioned that the user has to be trained to properly use BPMN 2.0 since all the possible constructs can be confusing. Petri nets analysis found inadequate support for modeling multiple instances, such as numerous start events can be identified. The organizational appropriateness merely points out the old tools using BPMN have language deficiencies, which can be solved with the new BPMN 2.0. However, empirical studies show that users work around these deficiencies illustrated by the analytical evaluation (Aagesen & Krogstie, 2015).

2.4.4 Business Process Management Notation 2.0

After processes are identified and discovered, they can be modeled and represented as a BPMN 2.0, which is a graphical notation in the form of a diagram. It provides an intuitive notation used by both technical and business users to represent complicated end-to-end process semantics, especially because many tools support this notation (Aagesen & Krogstie, 2015).

Further, BPMN 2.0 consists of 3 main diagrams, called the Business Process Diagram (BPD), the choreography diagram, and the conversation diagram. The BPD, which is the graphical notation can be divided into groups, pools, and lanes. Pools can represent a business entity or a business role, while a lane is a sub partition of a pool representing specific business roles

(Pérez-Castillo, & Piattini, 2013). Within these divisions, BPMN uses four different essential elements. Flow objects consist of events, activities, and gateways. They are joined with Connecting Objects, divided by a Swimlane, and enriched with information through Artefacts. Thereby, actives are the work that is being performed, which can be further broken down into processes, subprocesses, and tasks. Events happen without any action and can be start events, intermediate events, or end events. All elements of the BPMN Diagram can be seen in Figure 7.

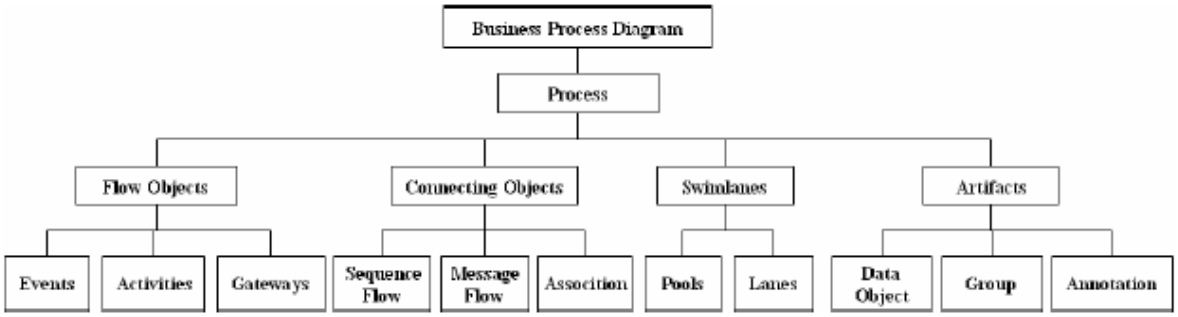

Figure 7 - BPMN Elements (Soo Kim et al., 2004)

When modeling these processes, Aagesen and Krogstie (2015) mention three different modeling levels. The first level is descriptive modeling, which is recording the process flow to understand As-Is and To-Be models. This is where BPMN is mainly being used. On the second level, analytical modeling, models become more accurate, allowing a qualitative, quantitative, and computer-assisted analysis for quality assurance purposes if used in the context of driving change in the organization. The executable modeling level produces XML-based specifications from the BPMN model to drive process engines enabling models to be activated automatically (Aagesen & Krogstie, 2015).

2.4.5 From BPMN to XPDL

XPDL is an XML-Process Definition language being used by a variety of workflow applications. Since BPMN is only a diagram, in order to be executable, it has to be translated into XPDL. For a better clarification Figure 8 shows the BPMN element on the left and the XPDL element on the right (White, 2003).

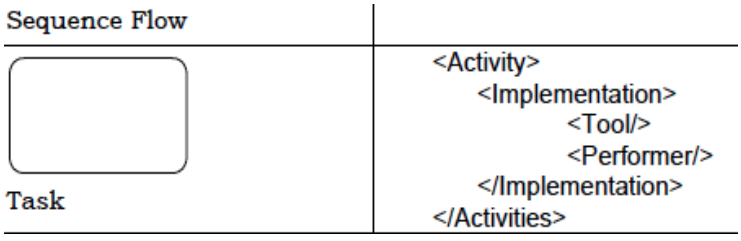

Figure 8 - Mapping from BPMN to XPDL (White, 2003)

Both BPMN and XPDL are flow-chart structures enabling a direct, simple, and complete translation from BPMN into the semantically identical and compatible XPDL. According to the WfMC (2019), XPDL writes out the BPMN diagram providing a file format. It enables other software to read and recreate the same process (WfMC, 2019). Depending on the need, BPMN diagram elements serve different purposes and thus are needed at different times. For example, Swimlanes, Artifacts, and the Connecting Objects Message Flow and Associations are not needed to execute the business process but add additional information to the BPMN Diagram. Further, Swimlanes give information about the role, who is performing the work. Artifacts and Associations add additional information to the model in written form. The Message Flow visualizes the flow of messages between entities without actually sending the information. Therefore, Figure 9 shows all the BPMN elements that have to be translated into XPDL (Jung et al., 2004).

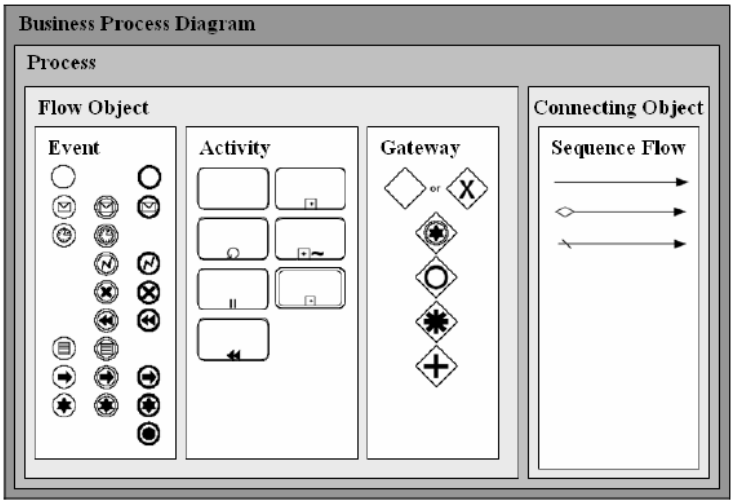

Figure 9 - BPMN Elements to be translated into XPDL (Jung et al., 2004)

As many elements serve as graphical representations, once BPMN has been translated into XPDL, the elements' order changes. Pool and Lane names graphically describe what each participant does in the form of containers. Thus, they hold information about activities, events, and gateways that are executed by the participant. However, these containers are not present in the XPDL format and cannot hold the participant information. Instead, the task level within the activity is the only entity that can hold the participant information. To associate the Pool or Lane with a task, XPDL utilizes the elements <*participant*> *and* <*performer*>, which carry the actor executing the activity. Therefore, each <*Task*> holds a hidden <*performer*> (Cheng et al, 2011).

2.4.6 Graph-Edit Distance

A distance measure has to be defined to calculate the graph similarity of two labeled graphs. Graph-edit distance (GED), first reported by Sanfeliu and Fu in 1983, is a widely accepted and used distance measure for labeled graphs due to its flexibility and sensitivity. The definition of GED is the minimum cost of an edit path between two graphs. The edit path is a sequence of

edit operations that change a graph to an isomorphic graph of B by adding, deleting, and substituting a node or an edge. These three operations are linked to a non-negative cost, which as a sum defines the cost of an edit path. Further, the edit costs transfer the metric properties to GED. For example, if graph A has real-valued nodes and edges with a calculated edit cost that uses the Euclidean distance, the GED is a metric on A (Blumenthal et al., 2020).

However, computing GED is complicated since it is NP-hard also if edit costs are uniform. Just using GED in matching algorithms would lead to computational explosions. Consequently, there needs to be a tradeoff between computational complexity and precision (Dijkman et al., 2009). As a result of the difficulty of computing GED or using approximation ratios to estimate GED, many new heuristics have been introduced in recent years. These new heuristics propose different approximation techniques of GED "via lower or upper bounds, using methods such as transformations to the linear sum assignment problem with error-correction, linear programming, and local search" (Blumenthal et al., 2020, p. 421). Nevertheless, only a few algorithms can handle large graphs that would allow, in principle, to design a GED algorithm performing more accurately than currently possible (Blumenthal & Gamper, 2020). Abu-Aisheh et al. (2018) mentioned that there are currently no reliable algorithms to compute GED for big graphs having more than 16 nodes within an acceptable time frame. Thus, using pure GED is not enough for a similarity search of process models, given that real-life process models often exceed 20 nodes.

Therefore, heuristic algorithms have been proposed that are based on graph similarity. The difference between graph edit similarity and GED is that *"[GES] of two graphs is the maximum possible similarity induced by a mapping between graphs [while the] GED of two graphs is the minimal possible distance induced by some mapping* "(Dijkman et al., 2009, p. 53). Hence, when using GES in algorithms, they are evaluated based on average precision and time. The best performing algorithm using GES was a greedy algorithm that maps between a pair of process graphs. It establishes 1-to-1 correspondences between nodes in the compared process models so that one node is at most related to one other node in the other process (Dijkman et al., 2009).

2.5 Previous Related Work

Using CBR for problem-solving is used in a wide range of different areas. Pichler (2011) used CBR and supply chain event management (SCEM) to enable business process management system (BPMS) more flexibility, as it reacts case-based on occurring events. Pichler (2011) states the general problem that standardized processes function as the basis for the process execution, while the instances of the process have to be adjusted to the current events if needed, which is not impossible in this form. Therefore, SCEM is used to identify and manage events that occur during process execution. It monitors processes and traces the whole process down to its activities' progress and sends out an alert if there is a disturbance in the process, such as activities that do not finish on time. CBR is used to find similar past cases from

the CBR database, which could be applied to the current event. The process and its model where the event occurred function as a basis for CBR systems search.

BPMS, using a processing language, can be divided into design time that models and stores processes, and run time executing the previous modeled processes. It is impossible changing BPMS processes during run time, making it impossible for the normal process flow to react to every possible event. However, Pichler (2011) finds that a process diagnosis should enable process adaptation. If this is often the case for the same process, it should be extended. The event's attributes and the case have to be compared one by one to find out if the current event and any case in the CBR system are similar. The sum of all distances of all attributes, which can be weighted if needed, shows if two cases match.

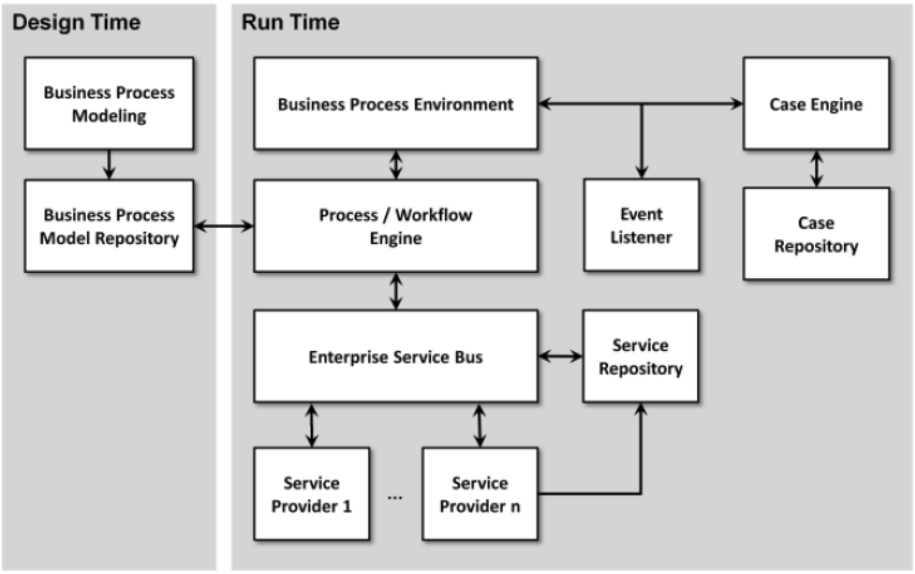

Figure 10 - Architecture of a BPMS with an Event Control- and CBR- system (Pichler, 2011)

Further, Pichler (2011) identified different categories of events that can be repaired, rescheduled, or replanned. If these three actions are not possible, past events have to be used to learn by drawing conclusions from recurring problems. In order to use these new methods, they have to be incorporated into the BPMS (see Figure 10). They are translated into an Event Lister and a Case Engine, which has a Case Repository. The Event Lister controls and monitors incoming new events and the processes affected. Identified events are reported to the process environment that decides how to proceed. If the standard process cannot handle the new event the Case Engine uses CBR and the Case Repository to find a suiting solution from a similar problem in the past. The solution from the past can either be automatically implemented or an administrator chooses from multiple possible solutions.

However, Pichler (2011) does not mention the possible language to be used. Neither was mentioned which language is most effective nor in what form cases and their solutions are being stored for later comparison. This process could be translated into BPMN to overview

better the standard process and where exactly problems occur. Further, a better analysis can be conducted on why the problem occurred (Aagesen & Krogstie, 2015).

The interaction between the BPMS and CBR – Engine is using the 4R – Model of CBR. The model is extended to a 6R – Model to satisfy the requirements of the process management. Hence the 6R – Model now consists of:

1. New Event
2. Retrieve Case
3. Reuse Case
4. Revise Solution
5. Reconfigure Process
6. Retain case

The current and the old case have to be compared to find a suitable case. In other words, the similarity between the cases has to be calculated. The higher the similarity, the more compatible are the two cases. Therefore, Pichler uses Petri Nets and a process description. The current case is characterized through the process instance, its execution status, and its description, which is then compared to the case base.

Although Pichler (2011) states the critical part needed to compare the old with new cases, it lacks an explanation of how each case's descriptions are being translated to be comparable, since different people use different semantics in describing a case or a process. Aagesen and Krogstie (2015) found that people working on process design tend to use text annotation extensively to work around design issues. Hence, much text would have to be analyzed instead of the process itself. Having a diagram in the first step to visualize and identify the process' problem would help to search for a solution with specific keywords in the case base.

One Petri Net consists of the process's statuses, its steps, and their sequences. A second Petri Net consists of the executing status and a description of the event and the process model. Everything is combined in a tuple:

- Identification of the process model from the repository on which the process instance is based
- Process instance's CEW-net with the status of the process, the steps in a process, and their sequences
- The current status of the CEW-Net with the current situation of the status of the process
- The occurring event that forces the change

Hence the characteristic part of the current event consists of three pieces: process models, execution statuses, and events, that have to be individually analyzed by the CBR system. Each of them is represented by a Petri Net, which functions are compared with one another. Once the similarity function found suitable past cases, their solutions can be adjusted. Pichler (2011) lists a variety of possible actions that can be done to adapt the process instance where the disruptive event occurred. They can be summarized in repairing activities and adapting processes of the process model. These activities have to be provided to the BPMS as a function to adapt process instances during the run time. The executed adaptation on the process instance is then saved as a solution in the case repository. However, once a suitable solution is found, there has to be a system to revise the solution to the current problem.

3 METHODOLOGY

3.1 DESIGN SCIENCE RESEARCH

To correctly apply Design Science Research (DSR), it is important to allocate the thesis in the right position on the knowledge contribution framework (see Figure 11). Considering that the thesis is researching a more efficient way to find the best solution for BPMN problems, it tries to combine many areas of knowledge into one system to improve an expert's decision-making.

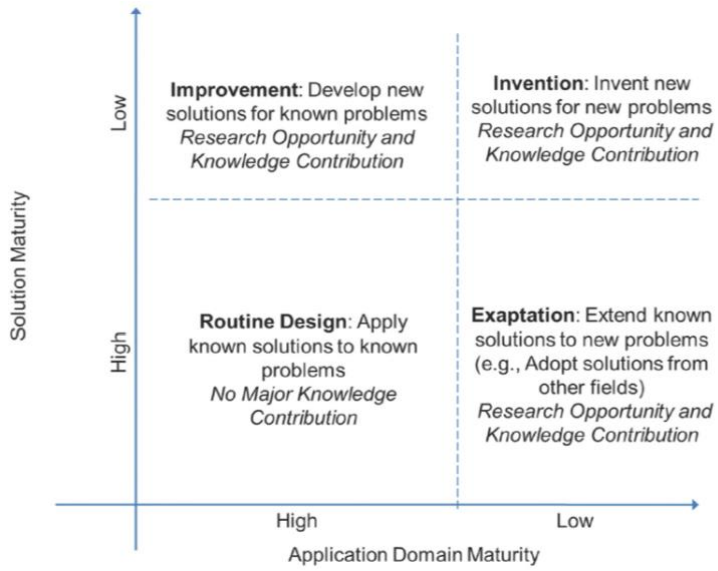

Figure 11 - DSR Knowledge Contribution Framework (Gregor & Hevner, 2013)

Therefore, in this thesis, the process to use past BPMN solutions to help solve a current BPMN problem can be seen as the "known problem". Hence, this thesis utilizes existing knowledge from different domains in an attempt to contribute to a possible new solution to the identified "known problem". Finding a new solution to a known problem offers a research opportunity and knowledge contribution while the solution maturity might be low, which positions this thesis in the quadrant of "Improvement".

DSR is the most appropriate method to design and develop artifacts, especially when the researchers' desired goal is an artifact in organizational and academic environments. These artifacts, which can be constructs, models, methods, and instantiations, are designed, reviewed, and justified for their importance to solve a problem. Hence, DSR focuses on problem-solving (Kanellis & Papadopoulos, 2009; Dresch, Lacerda & Antunes, 2015).

Further, DSR changes the state-of-the-world by introducing new and innovative artifacts built by the researcher's interests, values, and assumptions. The artifact's meaning is based on its functionality that it provides to the system (Kanellis & Papadopoulos, 2009). Moreover, DSR solutions eventually support improving existing theories and are a generalization of a specific class of problems helping researchers and practitioners. Hence, DSR is producing knowledge

to improve theories and is oriented towards a particular problem (Dresch et al., 2015). The DSR researcher's values are also subject to change as DSR is performed iteratively. New observations become the basis for a new theory that is being tested, beginning a new circle. This procedure could be seen as action research; however, DSR has a shorter time frame (Kanellis & Papadopoulos, 2009).

Therefore, the researcher in DSR is a pragmatist and must have a high tolerance for ambiguity as the research effort is not always apparent while being perceived as successful (Kanellis & Papadopoulos, 2009). Rather than focusing on the different assumptions of epistemology, ontology, and axiology under pragmatism, the researcher's priority is the underlying research problem and its research question leading to a practical outcome. Finding a solution, the researcher uses different kinds of views, knowledge, and methods to obtain a better picture implicating that there may be multiple realities (Saunders, Lewis, & Thornhill, 2019).

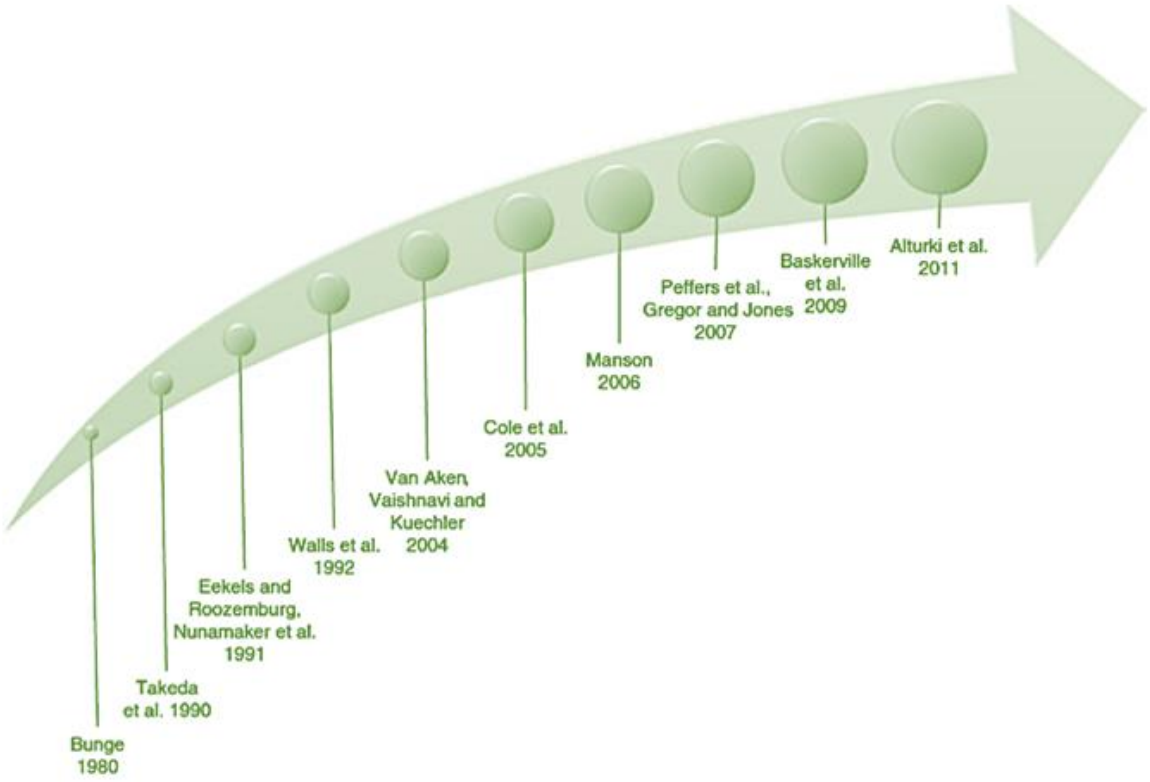

Figure 12 - Authors operationalizing design science (Dresch et al., 2015)

Organizations can use the results of DSR to solve practical problems. Therefore, a solid knowledge base needs to be built and utilized. In information systems, DSR develops practical knowledge for designing and implementing new information system initiatives, which can be positioned in theory-building and theory-testing. DSR methodologically constructs an artifact with an experimental proof – an experimental exploration of the theoretical method – and makes relationships between the artifact and its elements visible during the artifact's evaluation or construction phase. Thus, the most visible output of DSR is the artifact itself

(Kanellis & Papadopoulos, 2009). Dresch et al. (2015) show that there have been many authors before who formalized a DSR method from design science. The components authors in Figure 12 coincide when proposing a method of research based on design science. Based on these authors Dresch et al. (2015) could identify specific steps that need to be taken for a successful DSR implementation.

3.1.1 Identify Problem and Motivation

As a first step to develop an artifact in DSR, the researcher needs a concrete definition of the problem. Thus, one should start by defining a research question (Dresch et al., 2015). Therefore, one has to take into account the different possible research entry points (see Figure 13).

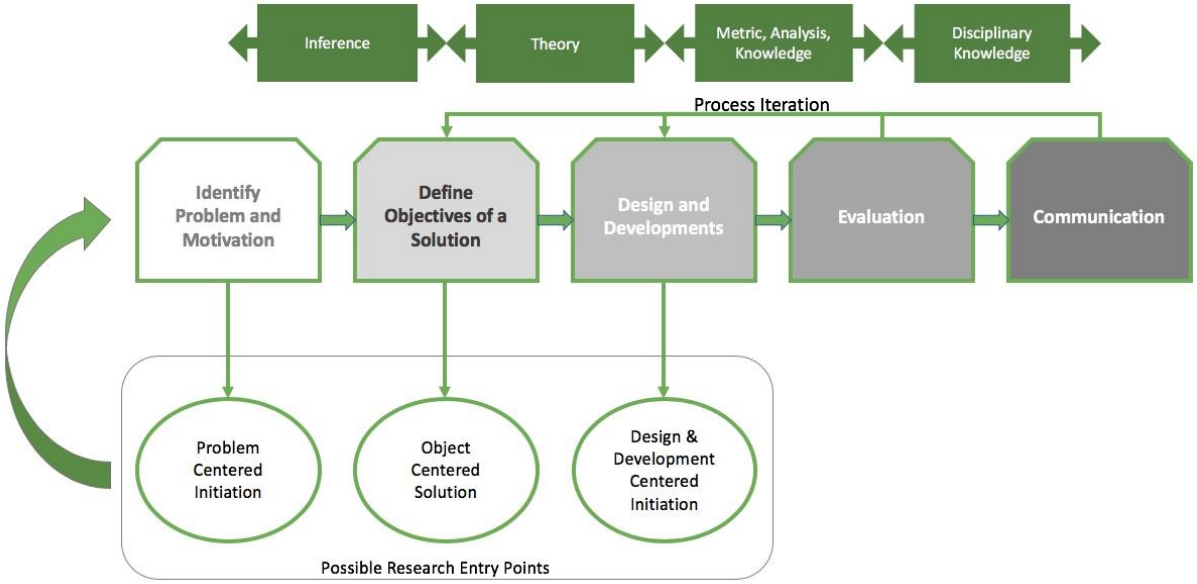

Figure 13 - Design Science Research Adaptation Methods

3.1.2 Define Objectives of a Solution

For the development of the artifact, the literature states that specific features and requirements need to be determined to have a solid base for developing a solution. This solution should be based on the problem definition and the limitations stating possibilities and what cannot be done (Peffers et al., 2007).

3.1.3 Design and Developments

To design and develop an artifact, one has to define the most appropriate state of the artifact to solve the underlying problem, which will be evaluated in the next step, demonstrating through research that is important to develop a solution. Conducting a literature review to review analyze theories in the field should serve as a basis for the research developed in DSR. With the acquired knowledge, it is possible to construct a solution according to the business needs and justify the proposed artifact's value and adequacy (Dresch et al., 2015 and Peffers et al., 2007).

3.1.4 Evaluation

The validity of the developed artifact's efficiency can be proven in an observational, analytical, experimental, testing or descriptive way, depending on the earlier determined problem and requirements. However, regardless of the decision method, the evaluation needs to be based on the previously defined business needs. Further, exploratory and confirmatory focus groups or interviews evaluate the performed work to achieve improvements during the artifact's development and demonstrate its usability when applied in the field (Dresch et al., 2015). The evaluation phase's outcome should argue whether the artifact is ready to be communicated to the academic community or if improvements have to be made (Peffers et al., 2007).

3.1.5 Communication

The outcome should be reflected on findings and the disclosure of results (see Figure 14). Its limitation and the newly generated knowledge of this thesis should finally be communicated to the academic community to assist other parties in their particular environment. Further, this stage should also explain how the artifact was built and the evaluation process for its validation (Hevner et al., 2004).

3.2 IMPLEMENTATION STRATEGY

This section explains the implementation of the five phases in the thesis and the actions taken in each step. As previously stated, this thesis is an **objective-centered solution**, which is why it starts with the definition of objectives and a solution to effectively store, retrieve, and adapt BPMN solutions that fit best the underlying BPM problem using CBR. Therefore, under point 1.2, the thesis' objective has been defined with multiple sub-research questions to precisely study the areas and domains required to reach the solution's goal (see Figure 14).

In the **Design and Development** stage, research has been conducted in AI, analogical learning, case-based reasoning, business process management, XPDL, and GED. A broad spectrum of research areas was needed to find individual solutions to each stage of the CBR life cycle. Thus, this stage was divided into three parts storing, retrieving, and adaptation that later work as one whole theoretical artifact.

The last stage of this thesis was the **evaluation** of the output artifact by experts in the area. The interview was the chosen medium to gather feedback from the participants about the proposed artifact's usefulness. Since the thesis is a qualitative study, the evaluation's qualitative primary data is reasonable (Gill et al., 2008).

The semi-structured interviews allow an in-depth analysis of the model by the interviewee offering great potential for insights, which are sought after at this stage given the need for feedback. Further, an interview gives each interviewee enough time to speak and dive deep into a conversation explaining their opinions about the artifact and its usefulness. It is also

easier to manage and organize to talk to each interviewee than to bring multiple experts together at the same time (Saunders et al. 2009, pp 320-360). The interviews are scheduled for 30-45 minutes, starting with a presentation of the artifact (Gill et al., 2008).

The interviewees were Pedro Maia Malta – Professor at NOVA IMS in the area of Business – IT Alignment with the approaches of BPM as a framework; Isabel Machado Alexandre – Professor at the ISCTE – IUL in the area of Information Science and Technology; and Frederico Cruz Jesus – Professor at NOVA IMS in the area of BPM.

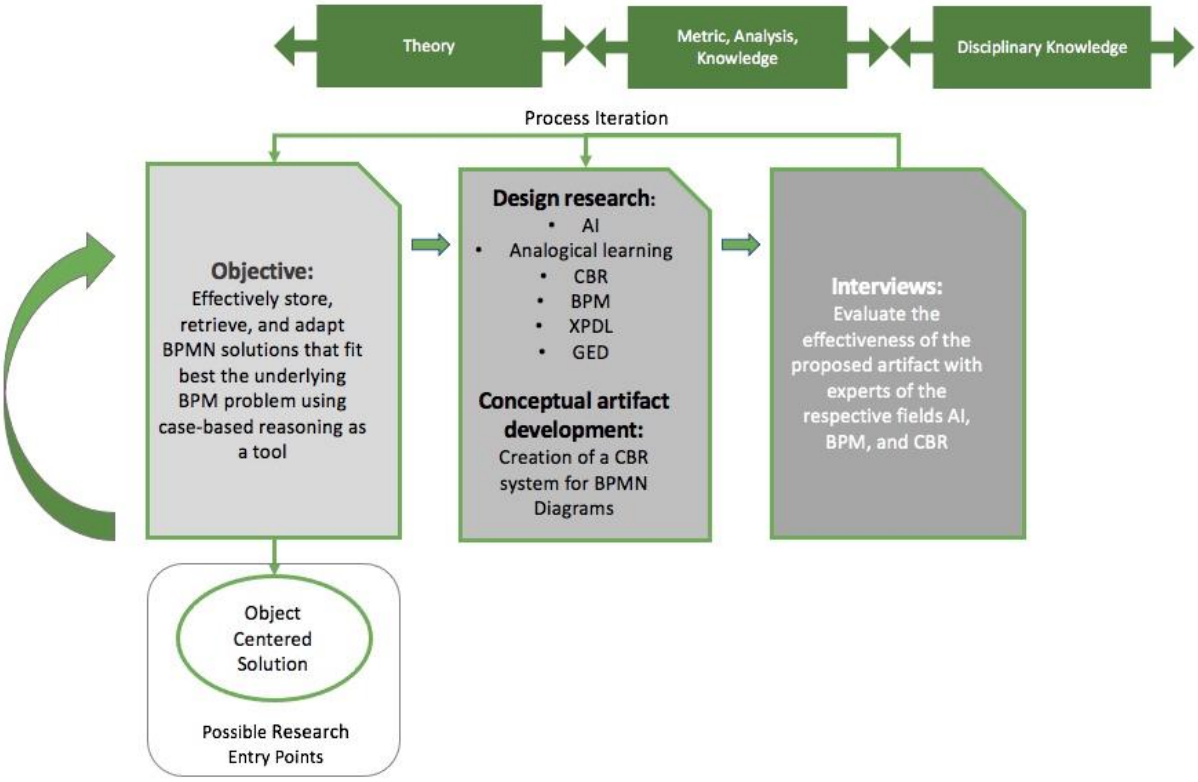

Figure 14 - Implementation Strategy

The **communication** step is not part of the thesis since it is out of scope due to the lack of time. However, future case studies are proposed to test the proposed artifact's technical implementation possibilities and understand its effectiveness and how it could be improved. The thesis has been published in a journal to present its usefulness and effectiveness to the academic world.

4 STORING BPMN DIAGRAMS

Using the CBR life cycle, the first cases have to be stored to build a case database that can be later used to optimize BPMN diagrams' problems (see Figure 15). Before any BPMN diagram can be stored first, the key information must be retrieved for proper labeling to index each diagram correctly. To be able to retrieve information, the BPMN diagram has to be first converted to an XPDL file, which keeps the diagram's structure. Thus, the indexing system is only working with the BPMN's respective XPDL files, which can later be translated back into a BPMN diagram. Therefore, only the XPDL file of each BPMN diagram has to be stored.

There are specific sections within XPDL that contain information needed to build the labels for the BPMN diagram's index. Creating the BPMN diagram index at the storing stage helps reduce the computing time later in the retrieving phase. Further, the BPMN information used is limited to the diagram itself. No additional information, such as activity time, is used to analyze this thesis for simplicity reasons.

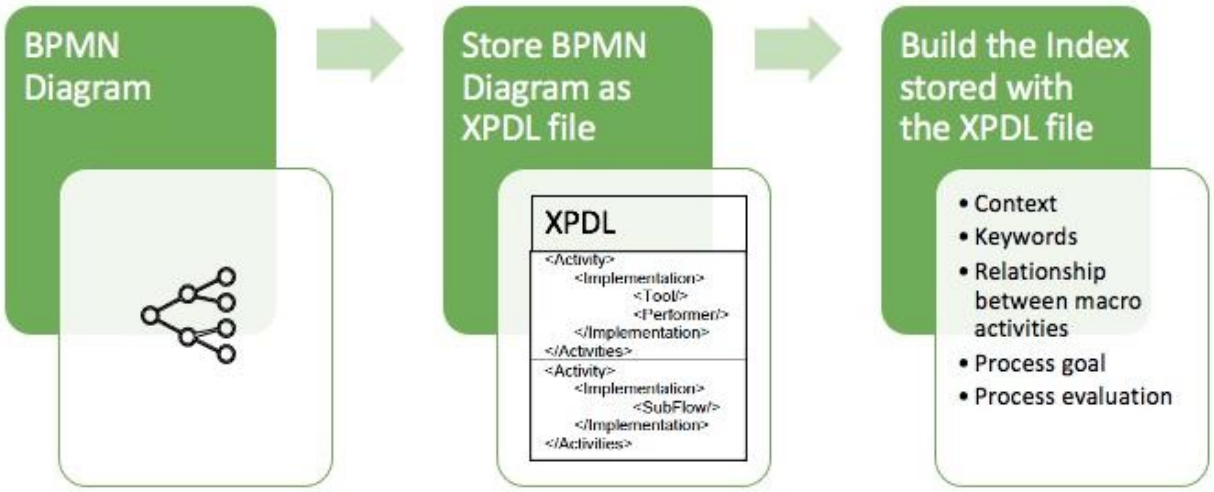

Figure 15 - From a BPMN Diagram to its Index

4.1 BUILDING THE INDEX

The index's labels are only effective if they adequately capture and describe the process content. Hence, each XPDL file's labels can be broken down into the following five different categories, namely context, keywords, the relationship between macro activities, process goal, and process evaluation. Together they capture the actors involved in the process, the tasks they fulfill, and how they relate to each other. Further, they contain the goal the process is supposed to achieve and its success or failure. This information will help later in the retrieving process to faster allocate processes with similar goals and use positive and negative outcomes to understand better how the right solution should look. Therefore, it strengthens the understanding of how processes are structured in detail and what professions are involved in getting to the desired goal.

4.1.1 Information in the original diagram

The context label consists of the process and participants' names for retrieving, which a few steps have to be taken. From where and what information is obtained from the XPDL file can be seen in Figure 16. First, the workflow process tag provides the name of the process. Next, as previously mentioned, BPMN's pool- and lane -names contain the information about the process participants, which are not directly represented within XPDL. Since only a task or activity can hold the participants information, the participants in XPDL associated with the pool and lane names of the BPMN diagram have to be retrieved from the activity itself. The context can show if a process needs different actors or if the current actor setup is a good fit for the current process problem. In the rest of this paper, the current problem refers to the problem case that is tried to be solved with the herein developed artifact.

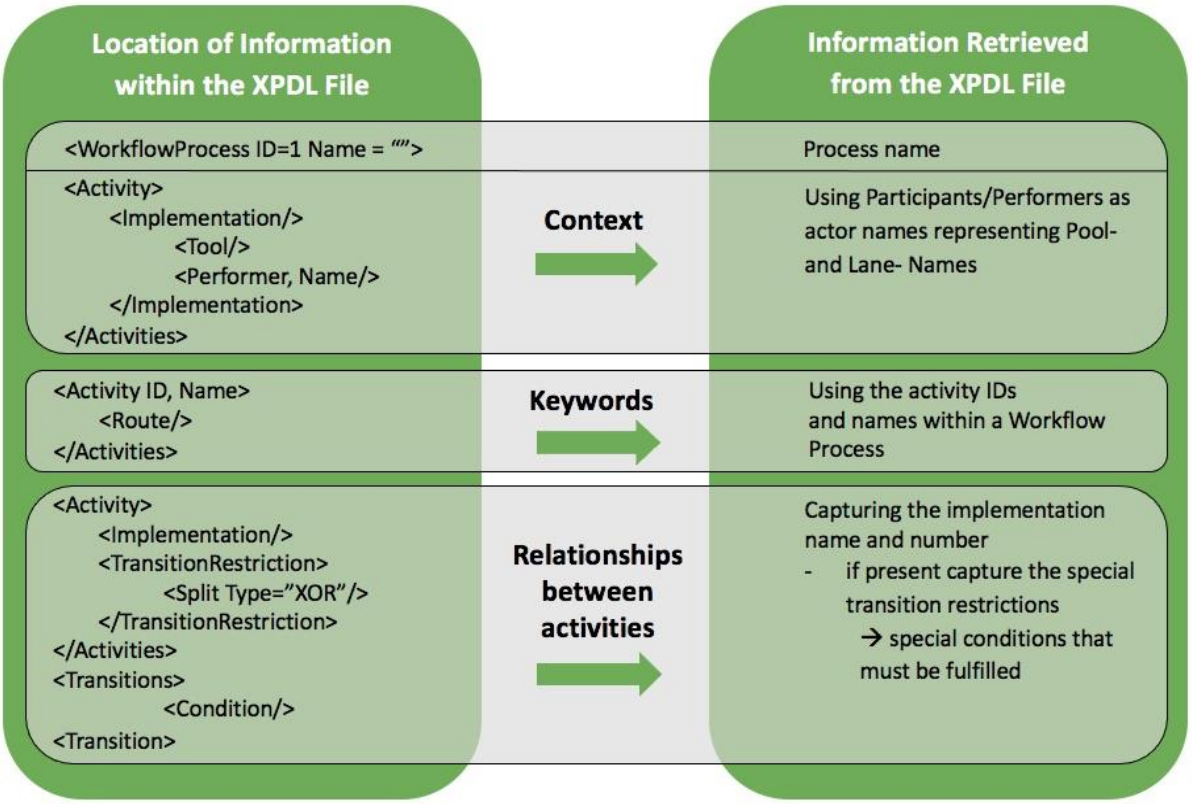

Figure 16 - XPDL Information Extraction

Further, the macro activities are used as keywords. Therefore, the keyword label summarizes the XPDL activity tags within a workflow process, which provides information about what happens during each activity. One process's activities can then be compared to other process's activities on similarity and if the current problem process might have to change, add or delete activities.

Next, the relations between macro activities have to be captured. Here the implementations, transitions, and conditions explain what happens with the token after each activity. In other words, as they are like the BPMN's gateway, they describe how two activities are connected,

for example, through an exclusive or parallel gateway. This gives additional information on how complex the flow of the diagram is structured. When compared to other processes, it can show the differences in structure and flow to reach the goal of the process. Although having the same goal, they can be vastly different and thus can show opportunities for improvement of the current problem process.

As a process can involve multiple participants, activities, or relationships, it is favorable to store them all together in a list. These lists can then be compared to each other position by position.

4.1.2 Information not in the original diagram

The following information is not directly translated from a BPMN diagram into XPDL; however, it has beneficial effects when added to XPDL as it makes comparisons easier. Knowing the goal of a process at the beginning will help later to compare processes better. Processes with the same or similar goal can be grouped and thus limit the search space for similar solutions.

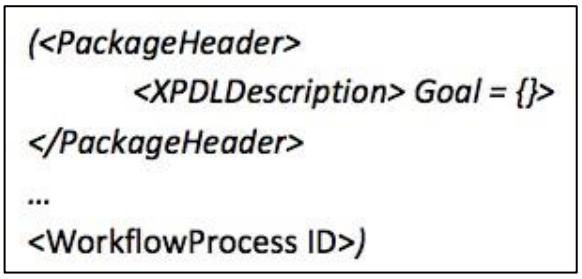

Figure 17 - XPDL Header adding the Goal Label

These similar solutions can the compared based on the previously explained labels context, keywords, and relationship between macro activities. Hence, if similar goals are available in the database, they can be an additional supportive label and can be added as a name into XPDL's package header (see Figure 17). Further, it is essential to have a process evaluation knowing whether the process was a success or failure (see Figure 18).

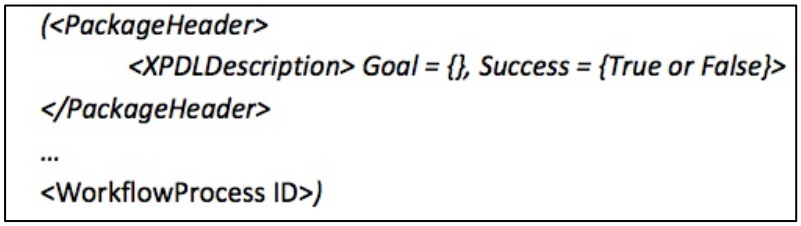

Figure 18 - XPDL Header adding the Success Label

Negative examples support building the new models by showing how the process goal could not have been reached. However, a more effective method is to separate the negative examples since this binary label would strongly polarize a later similarity search.

Another possible method is leaving successful and unsuccessful cases in the similarity search without having the current problem's success labeled, which would provide more freedom of search. Since the other cases in the case database would still possess this label, the other labels' similarity would determine the current problem's success or failure. Thus, there would be negative and positive examples retrieved, providing broader knowledge that would support a more accurate solution in the end. On the other hand, successful models possibly provide an instant alternative solution to the current problem.

Having established all five labels, they can be used to describe and classify each XPDL file process. Combined, they form the index for each BPMN process consisting of the process ID and the five labels. This construct can then be stored together with each XPDL file in the case database and be queried against the before-mentioned index labels. However, as the previous arguments show, there are many possibilities to combine and restrict labels during the similarity search to obtain one or multiple solutions to a current problem. These combinations and restrictions have to be chosen by the user's preferences and tailored to the knowledge required to find a solution to the current problem.

5 RETRIEVING BPMN DIAGRAMS

In the second phase of CBR matching cases to a current BPMN problem are being retrieved. Each XPDL file belonging to a BPMN diagram has been stored with its own index consisting of five different labels in the form of lists. The retrieving phase utilizes Carbonells suggested difference function in the form of a semantic search engine and a similarity metric to compare the lists with each other. The closest matches are then being returned for the adaptation step.

5.1 DIFFICULTIES OF RETRIEVING BPMN DIAGRAMS

Retrieving analogical BPMN diagrams means searching for some degree of similarity related to content or structure. The words and language each BPM specialist used can vary within the process schema, for example, naming the activities. Hence, two different BPM specialist can solve the same topic or problem using different words which are semantically similar. However, a normal search would only be able to find similar words with tokens or an edit-distance-based system. Therefore, semantics play a significant role when searching for similar BPMN diagrams.

Simultaneously, solutions in the case database may solve the same or a similar problem while having a completely different structure. In another scenario, the solutions may solve different problems with very similar structures. Thus, the structure is also important and can be different for each situation.

Consequently, two underlying problems have to be solved. The different diagram structures that could be beneficial and the used words that could be synonymous. While the diverse usage of words could mean the problem is covering a different problem, a different structure could just implicate another creative way of solving the current problem.

5.2 SEMANTIC SEARCH MODEL

A semantic search model queries the database's index for its keywords. There are many search engines that can retrieve information based on the similarity of words calculated by different edit-distances, but that is not the goal of this engine. Retrieving the same words limits the search to a strong match of similarly spelled words leaving just the possibility of a different diagram structure. This, however, would disregard the other powerful impact of semantically similar words. Therefore, the semantic search model uses a semantic engine that also finds semantically similar keywords or a list of keywords.

As previously established, the index information is obtained through the original XPDL file of the BPMN diagram (context, keywords, and the relationships of the activities) and some additional information (goals and success), which has to be cleaned first. A preprocessing stage has to take place, normalizing and tokenizing the data.

The semantic search model then uses the preprocessed index and calculates a semantic similarity score for each label of each diagram in the database. In order to calculate the similarity score, the semantic search model has to build a vector space, grouping together semantically similar words using constructs like unsupervised learning models. Afterward, the semantic search model calculates an overall similarity score based on the weight the user assigned to each label (different levels of importance for different labels) and returns the most similar solution. Hence, priorities have to be set on which labels are most important to the user.

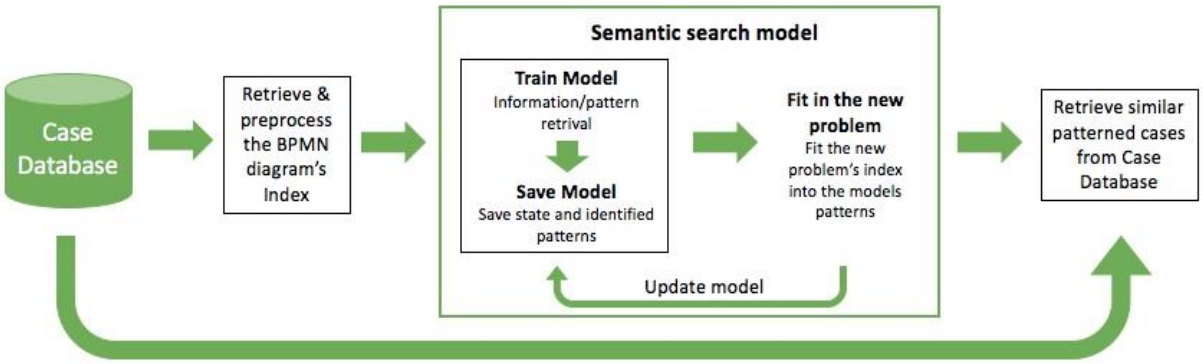

Figure 19 - Sematic Search Model used for retrieving similar cases

As a result, this paper proposes that the semantic model is trained by the whole database (see Figure 19). It breaks down the index and learns what phrases and word patterns, which are not necessarily linked to a specific label, exist in the database. Once a problem diagram is introduced to the model, it will compare the problem diagram's words to the database's pre-identified patterns. The space of patterns the new problem fits most can then be assumed to be semantically similar and retrieved. Hence, the model categorizes the new problem according to preidentified patterns. For example, there could be a pattern of how activities are being named around a particular process goal. Thus, the model remembers that these names are used when a particular process goal has to be achieved. If the words of the current problem are using words associated with this pattern, the model will retrieve the diagrams with the same word patterns from the database. As only word patterns are being analyzed, the index's context or goal labels can vary from the current problem giving a possible solution more freedom and more knowledge. Here the current problem index must not contain the label of the process' success. Otherwise, the semantic engine will bring the label into a context space with other processes sorted around the labels success or failure. This is not desirable when looking for knowledge that helps to find a successful solution to the current problem. The success label will help sort the most similar cases at a later stage.

It is important to notice that this analysis does not consider the order of activities in the diagram, as it just looks at the similarity of word patterns in the index, giving the search engine further freedom. In addition, the model's training time could be very time-consuming depending on the database's size; however, the training does not have to be repeated for

every search process because, without new cases, the patterns and phrases will not change. Instead, after a threshold of new diagrams has been stored on the database, the training process can be conducted during system downtime.

Nevertheless, the built index, which the model uses for the analysis, can be categorized as short text. Short text analysis has shortness and sparsity, which is a critical challenge for traditional text mining tools (Grida, Soliman, and Hassan, 2019). Traditional text mining tools require cohesive text to train the model to learn patterns and other information, which can later be found in other texts and documents. In the model's case, the patterns can be seen as semantic similarity clusters. Although much research studied the application of text mining tools for whole documents and more extended text parts, other papers point towards current research about short text analysis, for example, using discovering topic representative terms or self-teaching convolutional neural networks (Xu et al., 2017, Yang, Huang, and Cai, 2019). Hence, the semantic search model is still a theoretical construct, which practical development is out of scope for this thesis and is left to future research. Therefore, future tests are necessary to prove the viability of the above suggested construct and if the indexes are long enough and with adequate information to find and retrieve semantically similar indexes.

5.3 GRAPH-EDIT DISTANCE

Before, the semantic search model looked at the words used in different diagrams to find some similarities. This section looks at the BPMN diagrams' structure that visualizes the process flow from beginning to end. As previously established, the diagram structure holds much additional, important information. If a process is solved similarly to another process, it means that the structures coincide. The more the processes differ in the solution, so do their structures. Hence, when comparing two diagram structures for similarity, there should always be some threshold that allows for some structure differences providing some alternative structure solutions.

The need to compare two diagrams requires some metrics measuring their similarity, for which usually some distance has to be calculated. Therefore, this thesis proposes GED as a measure since the BPMN diagram's process structure can be translated into a graph (Dijkman et al., 2011). An example of the mapping from an XPDL to a graph can be seen in Figure 20.

Once the BPMN diagram is translated into a graph, its activities are now called vertexes, and its connecting objects are called edges. All other parts of the diagram are not considered in this step. GED can be applied to any graph where it calculates the minimum operations needed to adjust one graph to the other. Therefore, each vertex or edge from the first graph has to be mapped to one particular vertex/edge of the second graph or a dummy vertex/edge, which can absorb structural transformations.

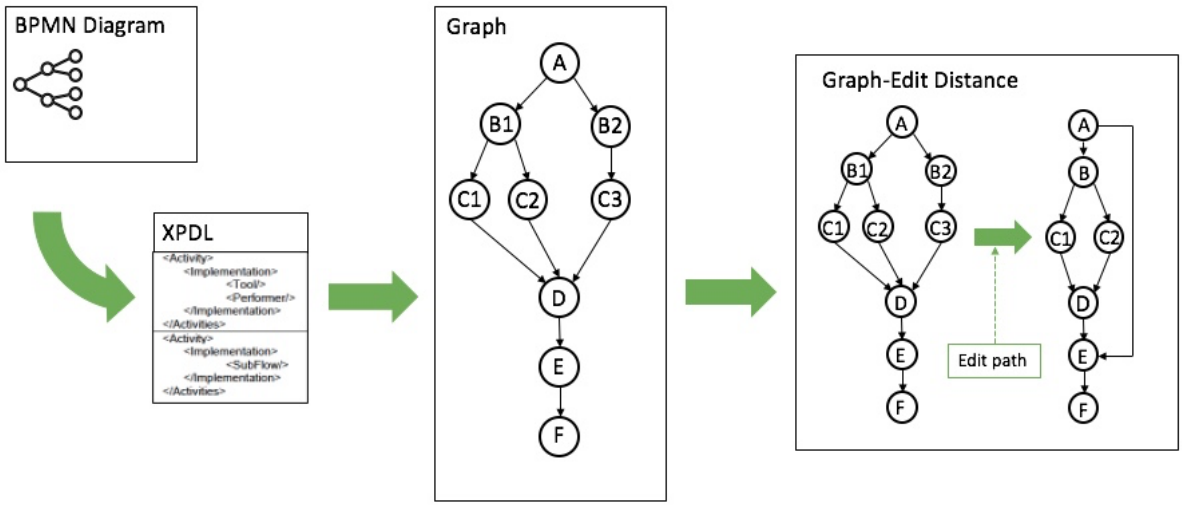

Figure 20 - From XPDL to Graph-Edit Distance

Changing one graph to the other is achieved through the edit operations insert, delete, and substitute, which sequence is called edit-path. This edit-path is applied to the graph's vertexes and edges. Next, the extent of the transformations applied to a graph by the edit-path is represented by a cost function. The cost function can measure the magnitude of distortions since each edit operation has a predefined cost. Hence, the edit-distance between two graphs is defined by the edit path with the lowest cost function indicating the highest structural similarity.

Since there is a lot of research on calculating graph similarity, there are already a few algorithms using the GES, a form of GED. According to the research, GES can be better handled by a greedy algorithm, which is currently the leading algorithm, especially when graphs get as big as a BPMN diagram with more than 18 nodes. However, the test and possible implementation of this GED form and algorithm is out of this thesis's scope. That is why this paper will use GED, which can later be translated into GES.

5.4 Retrieving using a Semantic Search Model and Graph-Edit Distance

The previous sections covered the retrieving phase analyzing the written and structural parts of a BPMN diagram. Now, as a final step, the retrieving phase has to coordinate these two analysis parts. First, both analyses need a threshold of the degree of similarity to retrieve similar cases, which has to be done by the user. The model needs the thresholds to find some similarity that narrows down the possible solutions because, without a degree of similarity, the system does not know what cases to retrieve. Once the thresholds have been set, the order of the two parts has to be set up.

Since semantics can better describe a solution's content and whether it has dealt with a similar or the same problem, the most logical order is for the semantical analysis to be followed by the structural analysis. Accordingly, the system must first learn the semantical context of

possible solutions to the problem, which it will then retrieve. After, it can look at the structures of the possible solutions with the same context. According to the threshold, based on the graph-edit distance or the GES and a greedy algorithm, it can then retrieve the solutions with the highest similarity. These solutions can then be used in the next step to adapt the current problem to the successful retrieved solution. Figure 21 shows how the two analyses can be merged to retrieve suitable solutions from the case database.

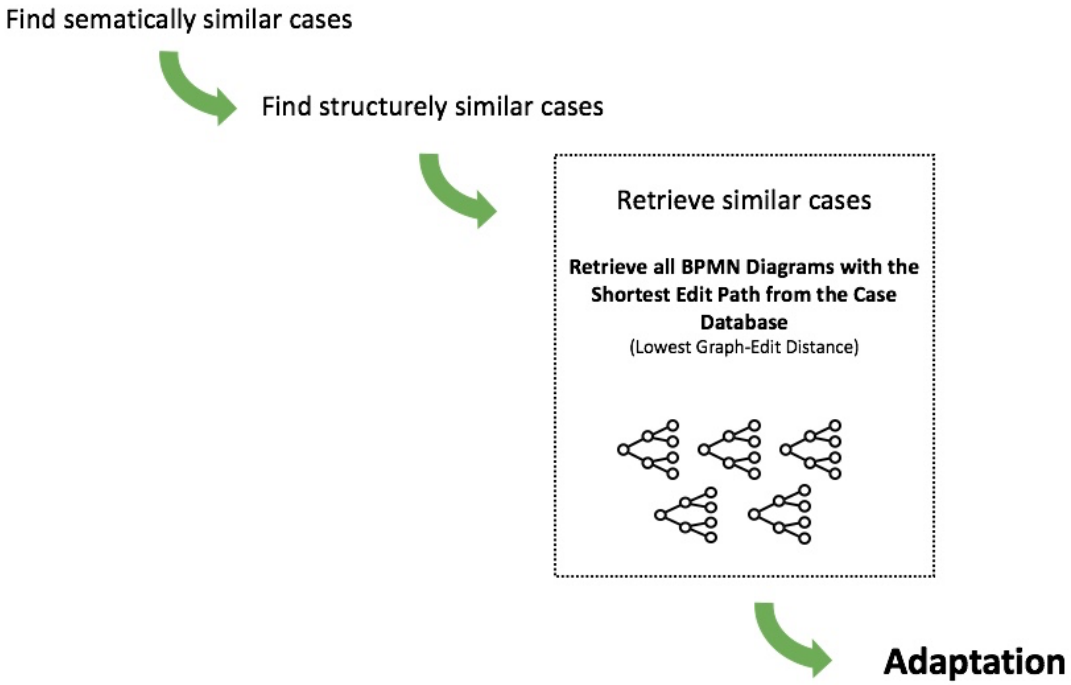

Figure 21 - From Semantical and Structural Analysis to Similar Case Retrieval

Retrieving possible solutions based on similarity always poses a knowledge barrier problem. As the system looks for some similarity, it is impossible to retrieve all knowledge present in the database. There might be a solution with no semantic or structural similarity to the current problem that could still offer a viable solution. This solution might even be from a completely different domain. However, since there is no similarity, the system will not be able to detect this kind of solution. Thus, these solutions also cannot be considered by the system.

6 ADAPTING AND RETAINING BPMN DIAGRAMS

The CBR live cycle's adaptation phase is about the knowledge transfer from the past solutions, retrieved from the case base to the current problem, which is not just about the similarity between the past solution and the current problem. According to Carbonell (1985), it is important to look at what kind of knowledge is transmitted from adapting the old solution to the current problem. Ideally, the adaptation model consists of positive and negative problem solutions, which share the same issues. Additionally, past solutions and the current problem should also have similar reasons of decisions taken available for a thorough understanding of the two problems at hand. If the reasons for decisions taken are congruent, the past solution could solve the current structure's problem.

In the current construct, the adoption phase can have positive and negative solutions available, the amount of which is based on the similarity threshold set in the retrieving phase. Additionally, the ratio of positive and negative examples can vary randomly for each CBR life cycle depending on the similarity score. However, not all information Carbonell suggests is available in the BPMN diagrams. For example, the reasons for decisions taken are often not written down in the diagram and thus are difficult to obtain. If information about the decisions taken were present in the diagram, it would be in the form of process descriptions used to explain some parts of the diagram. Since not all parts of the diagram are always explained, the reasons for decisions taken are often incomplete and would make the text analysis even more complicated and time-consuming. Hence, due to incompleteness and increased complexity of the retrieving phase, the reasons of decisions taken are not considered in this adapting approach.

In the end, there are only two points of knowledge available for the system to derive a valuable solution for the current problem, namely semantical context and a degree of similarity between the process models' segments. Consequently, there is not enough information to implement derivational analogy since the system cannot reuse decisions taken in the retrieved solutions and build a new solution from the ground up. Therefore, this report suggests a transformational analogy for the system to build the solutions to the current problem based on past solutions, which can deal with the context and structure of retrieved solutions and the current problem. Additionally, next to the transformational analogy, the adaptation system takes into account positive and negative solutions. Now that the basic concepts of the adaptation system are set, detailed procedures can be developed.

6.1 THE ADAPTATION SYSTEM

The adaptation system (see Figure 22) consists of multiple steps, from applying old knowledge to validating the new solution and its functionality. Since the semantic engine already retrieves the contextually most similar cases in the previous step and the GED distance further narrows down the search to the structurally closest solutions, all these solutions can be

temporarily stored in a separated pool (Step 1 of Figure 22). Next, the adaptation process (2) applies the changes (Figure 23). The GED has already calculated all changes that have to be made between the vertices and edges of the old solution and the current problem in the retrieving phase. In this case, the vertices are the process activities, and the edges are their connections. Thus, the adaptation process implements the most straightforward change calculated by the GED, making the current problem's process successful. In order to validate the successful change from the current problem to the new viable solution, the changes within the BPMN process's XPDL file have to be temporarily stored (3). Since the transformational analogy applies the best successful solution to the current problem, the negative examples have not been considered yet. Therefore, the negative retrieved solutions can be separated into an extra pool (4) so that in the next step, the temporary new solution can be compared to the negative examples (5).

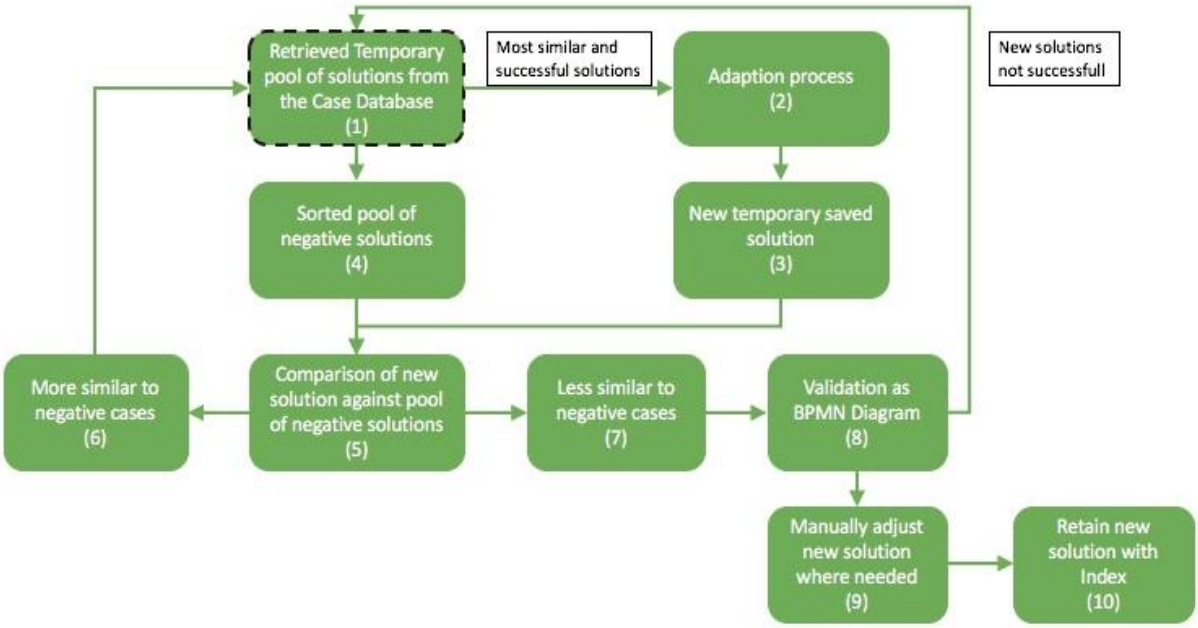

Figure 22 - Adaptation Process Overview

Since the system is now dealing with negative examples, the desired outcome is a low similarity between the new temporary solution and the case base's negative solutions. However, they will have some congruities, given all cases have been retrieved beforehand based on a high similarity score. Hence, the comparison looks at whether there is less similarity to the negative examples than before in both semantics and structure. It uses the previous similarity score that has been established during the retrieving phase and calculates a new overall similarity score.

The assumption is that if the old similarity score of the current problem and the negative solutions are similar in the beginning, the newly calculated score should be lower after the adaptation of a successful solution. Thus, according to the previous assumption, if the new solution is not less similar to the negative examples or if the new solution is even more similar

than before, it would indicate a negative, unsuccessful new solution (6). As a result, the next best retrieved successful solution would have to be retrieved from the temporary pool (1) to go through the adaptation and comparison process (2-5) again. Once the similarity is less than before (7), the assumption is that the new solution to the current problem must be more successful than before.

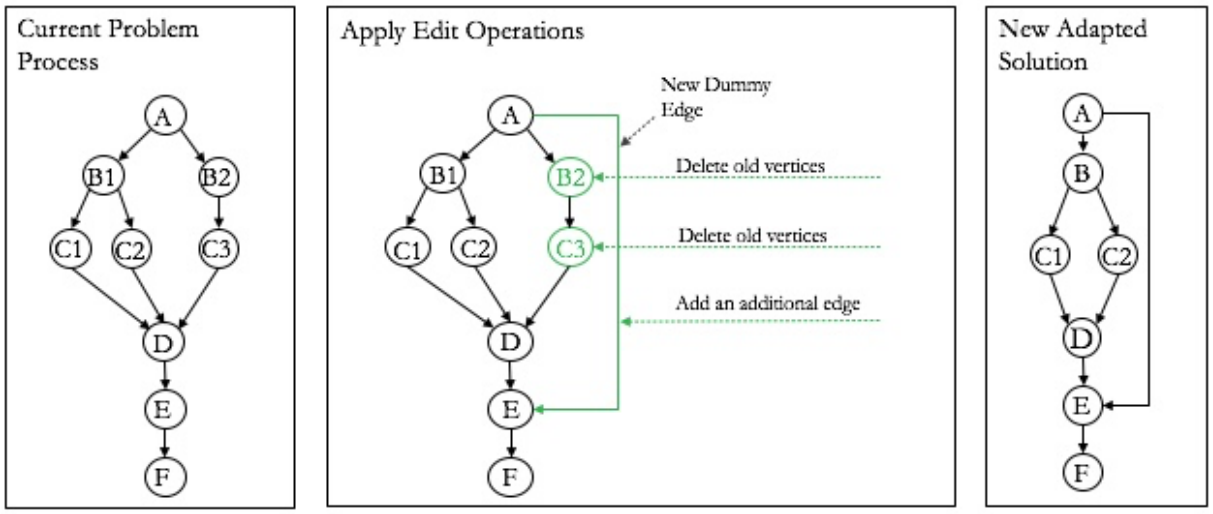

Figure 23 - Adaptation Process using Transformational Analogy

For validation purposes, the new solution, transformed into a BPMN Diagram, is then tested for a correctly running process (8). If it is also successful at this stage, the current problem can be solved with the new solution. However, this new solution was built based on the structure disregarding the context as it is assumed to be similar. Thus, there is the need to check for misinterpreted activities by the system, which contextually and structurally do not fit precisely the place the system put them in (9). For example, there is always the possibility of two times the same activity at different places in the process diagram. It can result from the two processes having the same activity before the adaptation but at different places. In addition, due to the double copy of one activity, there exists the possibility of a missing activity. In this case, it is required to manually adjust the new activities' order and language according to the current problem's context and compare the two diagrams' structure. Further, the user must also verify the process according to the needs and, if needed, adjust the process. Once the new problem solution is found viable, it can be retained in the case database as a new solution (10). Therefore, the new solution to the current problem is indexed as described in the section *storing BPMN diagrams*.

7 VALIDATION

For the validation, three interviews have been conducted with the three interviewees mentioned in 3.2 Implementation Strategy. They were Pedro Maia Malta (PM), Isabel Machado Alexandre (IA), and Frederico Cruz Jesus (FJ).

Each interview started with a presentation of the developed artifact, namely the CBR system solving current BPMN problems with past cases, which final solution is then retained again, growing the case database. Afterward, the participants gave their feedback, and the interviewer sought the answers to the three questions directly or indirectly. The three questions were the following:

1. Regarding the proposal of using the Case-Based Reasoning system, do you think the model is useful to solve BPMN problems?

2. Do you have anything to criticize about the proposed model?

3. Do you have any improvements and/or recommendations for the model?

Since the interviews were very rich in content, there was no direct answer or direct questioning. Thus, the validation section contains a summary of each interviewee's answers, which they have given at some point during the interview. The full transcripts of each interview can be found in the Appendix.

Question 1:

PM:
I think this is a good proposal. There is a good structure here, theoretical but a good structure. I think one good point is to retrieve the temporary pool with a number of examples, as the more solutions you have in the database, the more options you have. You will choose from a larger quantity of solutions, so you must do more correlations and thinking. As humans, we can do this during months and a computer, perhaps within a few minutes. If we have a data warehouse, a big data warehouse when we automate, and when we use technology, it is better for the performance as it is quicker, so the choice will be more oriented.

IA:
I think you identified the problems and the limitations, and if there is a good solution, it can definitely be found. I think CBR is a good way to deal with this because it is one of the advantages. However, the question is, the semantic that you extract from the diagrams can be limited. And as you said, the graphs can be complex. Sometimes, if you don't have the right cases in your database, it may be difficult to find the most similar one. So, it is always good to

wait for the validation from the human or the expert, whatever you want to call it. I think as it is the combination of semantic similarity and the graphical and graph structure, it's a good answer to your problem. To have a more precise solution, you have to have some cases and analyze what your model would get and what it doesn't get, which could also be useful.

FJ:
You have a sophisticated part that has to do with the research and how you search for previous models, which is more an area of data science than BPMN. I think this is interesting. However, this is then via underlying rudimental companies, right. The same method, the same process is already then in knowledge management, am I wrong? So, for example, in consultancy companies that have knowledge management, the idea is trying to get the same objectives as you propose here, which is not to reinvent the wheel in one sentence, but it's done in a more old-fashioned repository. Here, what you're saying is okay, so every consultancy or BPM project aims to improve the way companies do the business that is their work, right. Because we have BPM and models, we can enhance the knowledge management part by using the BPM and models to search for similar projects in the past, and you will not every time you have a project start from scratch. It seems very nice.

Question 2:

PM:
I would like to see some examples or some data tests.

IA:
The diagram that you have now on the screen is complex to follow—even the phases' steps 1, 2, 3, 4. Now at five, it can go bidirectional. So, it might not be complicated to follow for you. It's simple, of course, because you have developed it. However, for the rest, looking at it the first time, might have difficulties.

FJ:
I think a lot is missing, and you are generalizing. For example, this is the same process as some other case in the past, it can be the same process, but the distributions and the parameters of the processes, the arrival of new cases is completely different that will completely ruin your simulation. The real important part of quantitative analysis is the simulation. You really want to do a simulation, and you can only run a simulation if you fit the models with all the data you need regarding the activities, the decisions, the arrival rates, the resources, and your salaries. You see. And I think this is very hard to do because of CBR because there each case is literally a case. So, I would make it clear this is useful for a specific part of BPM, which is qualitative analysis.

Question 3:

PM:
When we delete, and we don't use the information anymore, and perhaps we don't really have a good use for every information that we had since the beginning. However, I don't think we should 100% forget what the other visions said. I always think we should consider something from everyone. Even when you validate this BPMN diagram and even if you deleted paths because you have a new solution, the deleted part could be like a backup database. I would advise you to be aware about these visions that we normally forget because we have a better model. I wouldn't forget the other information. I would take care in every step of what we are doing with some kind of validation. So, you can go further with a reason with someone or something verifying what we do. That would be good; for instance, we retrieve a temporary pool of solutions from the case database when you go to the sorted pool of negative solutions. Why are the temporary retrieved pools the bests or good data for this? Can they be sorted as negative or not? There is always a question between the steps to question. In BPM improvement, all the research talks about AS-IS the situation they found in the enterprise, and then let's work to a TO-BE that is, for instance, quicker, so let's try to use a simulation too within a BPMN editor and try to reorganize times and tasks and number of requests so we can redesign the process with that focus. So, they are not really validation issues, but they are reasons that justify your model and your steps.

IA
When you ask me what can be improved, I think it is always best to find an example and try to use it in your model. I know that you said your goal is to develop this model and to propose it in in your masters; it would be richer if you could find a company and someone that has some of the models and try to use and see how the model would behave when trying to find something similar cases. So, this is my only suggestion would be to try to get one example. And try to say, for example, this diagram, how it would work, how it would be the transition between those cases.

FJ:
My only concern is looking through BPM lenses, and there are two things to keep in mind: the models are much more than the elements. The BPMN models are a means to an end; they are not the means by themselves. A model is much more than the elements. So also, the metadata that comes with them, for each activity, right. And that is the most difficult part to get for CBR, and the most difficult part, from what I understand, to put in that file system you are using.
So, what I eventually suggest is that along with the model, when someone finds a similar project in the past, you say that everything else goes along with that project. The reports, then the qualitative analysis, the quantitative analyses, the description of the case, and the process analysis's thoughts to improve the business process because you need a picture of what was the best project. I want the AS-IS and eventually the report with suspending the analysis.

That's what I want and what eventually is your choice for the most appropriate TO-BE will be the AS-IS that is more similar to what you have. I think they add more value. The models by themselves are not useful enough. If you give me a process model of a accompany similar to the one, I am doing BPM project for, the models will be almost useless. Even the TO BE because for me to see what is the TO BE, I need to see what the original starting point was. And I need to have the justification for what changed in that case because not everything will adapt to my ongoing project. Think about the laws, regulatory pressures, and contexts that companies need to follow in the USA but are not applied in Portugal. If you don't have all of this with you, and you focus excessively on the model, I think you might be missing a big point.

8 DISCUSSION

This section analyzes the developed artifacts' usefulness, critique, and possible improvements based on the validation section's answers. The outcome will be a general evaluation of the proposed model.

All interviewees agreed to the usefulness of the theoretical model and its learning from past cases. Although they agreed to the usefulness, they gave feedback to different parts of the model. It was seen positively that the model would work faster in finding old, similar cases than humans improving the knowledge management's performance. Especially, the pool of retrieved, similar solutions could lead to a more oriented outcome. Although some concerns were expressed that there can still be no fitting solution, the model retrieves the most similar solutions based on a similarity count system, so a solution is always returned. Hence, there should also be a person checking the new process's viability, which is also part of the model. Further, the interviewees mentioned a certain similarity between the model and its use in consultancy companies on a basic level, comparing it to their knowledge management learning from old cases since they do not always start from scratch for every project. These statements show that learning from past cases has positive aspects and is already used in some professional environments, confirming that CBR used in the developed model is useful.

However, there has also been criticism of the model of different aspects. Some interviewees would have preferred more practical examples and data tests, such as company data, which goes beyond the thesis's scope. As mentioned in previous sections, the purpose of this thesis is to develop a theoretical model. If the model is deemed valid, and useful future research involves its practical implications. Further, it was argued that the BPMN diagram alone is too general to draw similarities and does not provide enough data. Many other parts of the model influence the process's simulation, which is the most crucial part of validating the new process. With this additional information, it is tough for CBR to analyze because every case is very individual. So, if CBR was used, it has to be focused on qualitative analysis. As the beginning of the thesis mentioned, it has only been focused on the BPMN Diagram for simplicity of the theoretical model.

The recommendations focused mainly on the BPMN details used to find similar cases and to adapt them afterward. If one deletes information and parts of the processes 100 %, they are not available anymore if any other problems arise with the process where the deleted parts could be useful. It is never practical to delete information 100%. Instead, there should be a backup database that stores the deleted parts. As a result, a backup database and practical tests in the adaptation stage should be considered. While the backup database can be incorporated into the adaptation stage, the technical implications are future research.

Further, there should be some kind of validation in every step that verifies why one moves forward. The model's validation steps have been considered, primarily through the final human validation step of the new process's correctness. As mentioned under criticism, just searching for the similarity between the BPMN diagrams is not enough since more detailed information is important to make the process work and verify it through simulations. Thus, it was recommended to continue to find similarities between the BPMN diagrams as proposed. However, instead of just storing and retrieving the diagram, once a similar old diagram has been found, all its additional information should be retrieved with it in order to properly understand the old solution to adapt it correctly to the current problem. Therefore, even the old AS-IS and TO-BE models would be essential to understand the justification. This could be added to the current theoretical model increasing the human validation step's importance.

All in all, the theoretical model itself has been evaluated to be useful, although it requires adjustments for additional BPMN information in the adaptation part. It became evident that the human factor in checking and analyzing the new and final process through simulation is significant for its validation. Hence, these points will have to be considered for the theoretical model's practical implications.

9 CONCLUSION

Based on extensive research in BPM and CBR, together with side topics, such as GED, it can be concluded that a whole CBR life cycle can be applied to BPMN diagram problems with the need for human intervention.

As BPM at its core controls and optimizes the business processes to make them more effective and efficient, it ultimately increases shareholder value. CBR, as part of AI, tries to mimic human learning by analogy. Hence, CBR is adopted to support BPM, making better decisions with existing knowledge when solving process problems. Therefore, BPMN diagrams must be translated first into XPDL files before CBR concepts and transformational analogy can be applied. Using the CBR life cycle helped to structure the solution's steps to build a closed learning process.

During the storing phase, each XPDL file obtains an index with five labels that contain the needed information for efficient comparisons between old solutions and the current problem later on. After the old solutions have been stored in the case database, the retrieving phase starts with semantically analyzing the current problems' index. Further, it will place the current problem in the semantic space with the old solutions that use similar wording, which is expected when dealing with similar solutions. The old solutions the semantic analysis finds closest to the current problem are then extracted for structural analysis. Here the XPDL structure is translated into a graph to apply GED distance to calculate the smallest edit distance to adjust the old solution to the current problem. Thus, the smallest GED distance indicates the final most similar solution. In the final adaptation phase, the transformational analogy is used, fundamentally applying the smallest GED to the current problem. Afterward, the new structure of the current problem has to be tested for its viability. If the user finds the current problem’s solution successful, its wording must be adjusted before it is stored in the case database starting a new CBR cycle. While the proposed artifact's usefulness is generally approved, it needs improvement in some areas.

This dissertation was developed with the purpose to contribute to the development of learning from past solutions by building an overall system that stores, retrieves, and adapts old BPMN solutions to current BPMN processes. Therefore, this work did not have the objective to solve the whole problem but to contribute to a possible solution by using CBR. Combined with BPMN diagrams each CBR phase needs a different and unique approach to bring the whole CBR life cycle together as one working solution. The solution is constructed on a theoretical level and thus does not provide a practical implementation.

9.1 LIMITATIONS OF THE PROPOSAL

First of all, the validation group contained only a few experts limiting the shared knowledge. Nevertheless, individual interviews minimized this constrained since it offered more insights and personal opinions from the interviewees, that were selected experts for different parts of the model.

Secondly, this research focused on building a theoretical model; it did not develop the technical side because it is outside the thesis's scope due to time constraints. The unique human brain quickly draws connections between BPMN cases, which is challenging to do for a machine. Therefore, the developed semantic search engine is a theoretical construct with not yet reached technological requirements. Until now, most text mining tools for natural language processing need excessive amounts of concise text to be trained and to conduct an analysis. Further, most search engines look for the exact word or phrases of words on other documents limiting the retrieval of other matching knowledge.

Moreover, big BPMN diagrams can still be too big for similarities to be calculated by GED or GES in an A* algorithm. Lastly, adaptation is a challenging part as it is limited to blindly applying the structure without checking if the content of each activity also fits in the same space. There are still many problems in the adaptation phase, so the user must intervene to check the outcome, maybe even for the temporary outcomes.

9.2 FUTURE WORK

As future work, a bigger group of experts should evaluate the theoretical model. It would consolidate the discussion about the model's improvements, emphasizing its strengths and weaknesses, which can then be considered before advancing with technical implications.

Further, the proposed model is limited to the diagram's information, not considering all the additional metadata that describes the diagram's process in detail, such as the processes descriptions and activity times. However, the model theoretically has the ability to save the additional information and retrieve it together with the BPMN diagram if matched with a current similar BPMN problem. Additionally, deleted sequences in the adaptation phase should be stored in a backup database for possible later use. Future research has to consider these two additional information points before it advances on the technical side.

From the technical side, further research needs to determine how effective semantical search for keywords or key phrases broadens the retrieval of possible applicable knowledge. Therefore, the semantical model's and GED's capabilities have to be tested on practical examples to confirm their viability and usefulness.

Hence, future research should extend the model's theoretical possibilities and its technical implications, which will minimize the need for user intervention.

BIBLIOGRAPHY

Aagesen, G., & Krogstie, J. (2015). Bpmn 2.0 for modeling business processes. In Handbook on Business Process Management 1: Introduction, Methods, and Information Systems (pp. 219–250). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-45100-3_10

Aamodt, A., & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches (Vol. 7). AI Communications. IOS Press. Retrieved from http://www.iiia.csic.es/~enric/papers/AICom.pdf

Abu-Aisheh, Z., Raveaux, R., Ramel, J. Y., & Martineau, P. (2018). A parallel graph edit distance algorithm. Expert Systems with Applications, 94, 41–57. https://doi.org/10.1016/j.eswa.2017.10.043

Atay, M., Chebotko, A., Lu, S., & Fotouhi, F. (2007). XML-to-SQL Query Mapping in the Presence of Multi-valued Schema Mappings and Recursive XML Schemas. 603–616. https://doi.org/10.1007/978-3-540-74469-6_5

Blumenthal, D. B., Boria, N., Gamper, J., Bougleux, S., & Brun, L. (2020). Comparing heuristics for graph edit distance computation. VLDB Journal, 29(1), 419–458. https://doi.org/10.1007/s00778-019-00544-1

Blumenthal, D. B., & Gamper, J. (2020). On the exact computation of the graph edit distance. Pattern Recognition Letters, 134, 46–57. https://doi.org/10.1016/j.patrec.2018.05.002

Carbonell, J. G. (1985). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition (No. CMU-CS-85-115). Carnegie-Mellon University Pittsburgh PA Dept of Computer Science. Retrieved from http://repository.cmu.edu/compsci

Cheng, R., Sadiq, S., & Indulska, M. (2011). Framework for business process and rule integration: A case of BPMN and SBVR. Lecture Notes in Business Information Processing, 87 LNBIP, 13–24. https://doi.org/10.1007/978-3-642-21863-7_2

Corchado, J. M., & Lees, B. (2001). Adaptation of Cases for Case Based Forecasting with Neural Network Support. In Soft Computing in Case Based Reasoning (pp. 293–319). London: Springer London. https://doi.org/10.1007/978-1-4471-0687-6_13

Dijkman, R., Dumas, M., Van Dongen, B., Krik, R., & Mendling, J. (2011). Similarity of business process models: Metrics and evaluation. Information Systems, 36(2), 498–516. https://doi.org/10.1016/j.is.2010.09.006

Dijkman, R., Dumas, M., & García-Bañuelos, L. (2009). Graph matching algorithms for business process model similarity search. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5701 LNCS, 48–63. https://doi.org/10.1007/978-3-642-03848-8_5

Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). Design Science Research. In Design Science Research (pp. 67–102). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07374-3_4

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). Fundamentals of Business Process Management. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33143-5

El-Sappagh, S. H., & Elmogy, M. (2015). Case Based Reasoning: Case Representation Methodologies. (IJACSA) International Journal of Advanced Computer Science and Applications (Vol. 6). Retrieved from www.ijacsa.thesai.org

Grida, M., Soliman, H., & Hassan, M. (2019). Short text mining: State of the art and research opportunities. In Journal of Computer Science (Vol. 15, Issue 10, pp. 1450–1460). Science Publications. https://doi.org/10.3844/jcssp.2019.1450.1460

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. MIS Quarterly: Management Information Systems. University of Minnesota. https://doi.org/10.25300/MISQ/2013/37.2.01

Gill, P., Stewart, K., Treasure, E., & Chadwick, B. (2008). Methods of data collection in qualitative research: interviews and focus groups. British Dental Journal, 204(6), 291–295. https://doi.org/10.1038/bdj.2008.192

Hammer, M. (2015). What is business process management? In Handbook on Business Process Management 1: Introduction, Methods, and Information Systems (pp. 3–16). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-45100-3_1

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly: Management Information Systems, 28(1), 75–105. https://doi.org/10.2307/25148625

Jung, M., Kim, H., Jo, M. H., Tak, K., Cha, H., & Son, J. (2004). Mapping from BPMN-Formed Business Processes to XPDL Business Processes. 422–427. Hanyang, Korea: Department of Computer Science and Engineering, Hanyang University. Retrieved from https://pdfs.semanticscholar.org/ef2c/a80a6826d8a394158c71103862f2b71b30da.pdf

Kanellis, P., & Papadopoulos, T. (2009). Conducting Research in Information Systems. In Information Systems Research Methods, Epistemology, and Applications (pp. 1–34). IGI Global. https://doi.org/10.4018/978-1-60566-040-0.ch001

Kolodner, J. (1995). Promoting Transfer through Case-Based Reasoning: Rituals and Practices in the Learning by Design Classroom and Evidence of Transfer. San Mateo, CA: Morgan Kaufmann. https://www.researchgate.net/publication/250616847_Promoting_Transfer_through_CaseBased_Reasoning_Rituals_and_Practices_in_the_Learning_by_Design_Classroom_and_Evidence_of_Transfer/citation/download

Kuchibatla, V., & Muñoz-Avila, H. (2006). An analysis on transformational analogy: General framework and complexity. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4106 LNAI, 458–473. https://doi.org/10.1007/11805816_34

Pantic, M. (2006). Introduction to Machine Learning and Case-Based Reasoning. [Syllabus] Retrieved from https://www.semanticscholar.org/paper/Introduction-to-Machine-Learning-%26-Case-Based-Pantic/1b926ea42ae3bf9e41ce1d447ca7f8ad1b0b7d6b.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45-77. Retrieved from http://www.tuunanen.fi.

Pereira, J. L., & Silva, D. (2016). Business process modeling languages: A comparative framework. In Advances in Intelligent Systems and Computing (Vol. 444, pp. 619–628). Springer Verlag. https://doi.org/10.1007/978-3-319-31232-3_58

Pérez-Castillo, R., & Piattini, M. G. (2013). Uncovering essential software artifacts through business process archeology. In Uncovering Essential Software Artifacts through Business Process Archeology. IGI Global. https://doi.org/10.4018/978-1-4666-4667-4

Pichler, A. (2011). Flexibilität in Business Process Management Systemen durch Case-based Reasoning. Association for Information Systems AIS Electronic Library (AISeL). Retrieved from http://aisel.aisnet.org/wi2011

Rosemann, M., & vom Brocke, J. (2010). The Six Core Elements of Business Process Management. In Handbook on Business Process Management 1 (pp. 107–122). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-00416-2_5

Sanfeliu, A., & Fu, K. S. (1983). A Distance Measure Between Attributed Relational Graphs for Pattern Recognition. IEEE Transactions on Systems, Man and Cybernetics, SMC-13(3), 353–362. https://doi.org/10.1109/TSMC.1983.6313167

Saunders, M., Lewis, P., & Thornhill, A. (2009). Research methods for business students. Pearson education.

Saunders, M., Lewis, P., & Thornhill, A. (2019). Chapter 4: Understanding research philosophy and approaches to theory development. Research Methods for Business Students.

Sciglar, P. (2018). What Is Artificial Intelligence? Understanding 3 Basic AI Concepts - Robotics Business Review. Retrieved June 26, 2019, from https://www.roboticsbusinessreview.com/ai/3-basic-ai-concepts-explain-artificial-intelligence/

Van Der Aalst, W. M. P., La Rosa, M., & Santoro, F. M. (2016). Business process management: Don't forget to improve the process! Business and Information Systems Engineering, 58(1), 1–6. https://doi.org/10.1007/s12599-015-0409-x

WfMC. (2019). About XPDL. Workflow Management Coalition. Retrieved June 27, 2019, from https://www.wfmc.org/standards/xpdl

White, S. A. (2003). XPDL and BPMN. In L. Fischer (Ed.), Workflow Handbook 2003 (10th ed., pp. 221–238). Future Strategies Inc. Retrieved from www.bpmi.org

Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., Zhao, J., & Xu, B. (2017). Self-Taught convolutional neural networks for short text clustering. Neural Networks, 88, 22–31. https://doi.org/10.1016/j.neunet.2016.12.008

Yang, S., Huang, G., & Cai, B. (2019). Discovering Topic Representative Terms for Short Text Clustering. IEEE Access, 7, 92037–92047. https://doi.org/10.1109/ACCESS.2019.2927345

APPENDIX

The transcripts in this section are for more detailed information about each of the three interviews. The wording of the transcripts has been adjusted to enable a clean reading of the conversations.

INTERVIEW 1 – PEDRO MAIA MALTA

Philipp Tueschen:
Hello, thank you, thank you for your time.

Pedro Malta:
Hello, yes of course no problem.

Philipp Tueschen:
So, my thesis is about CBR in BPMN Design. So, my focus was on effectively storing and retrieving BPMN solutions that best fit the underlying BPMN problem using CBR as a tool, so basically, I was building a CBR system that returns a past solution to a current problem. Therefore, I used the CBR life cycle as a structure, so I had to find a solution to each CBR stage – retrieve, reuse, revise, and retain. Starting with the BPMN diagrams, we have first to find an idea of how to store a BPMN diagram. My idea was to translate a BPMN diagram into an XPDL file, which is easily translated. We can then easily extract information about the BPMN diagram from the XPDL file to build an index that can then be stored with the XPDL file in the case database. This index can then be queried at a later stage.

Pedro Malta:
This index was built with some criteria. Did you find some criteria to build the index?

Philipp Tueschen:
Yes. I go first into the XPDL file. I extract the information I want to have for each part of the index. So, one label of the index would be context. There I look at the participants, the pool, lane names, what a person is executing, and the process's activities. Then I would store the context as one string, and at the end, I would have multiple strings in the index. The second label of the index would be the keywords, which are the activity names. They tell you what is happening in the process.

Pedro Malta:
The action itself.

Philipp Tueschen:
The action itself, exactly. Under the relationship between the activities, I store the connection between the activities, for example, the gateways. So basically, I would have all the information from the diagram translated into three separated labels. Then I add another two

labels to the index, which are later quite helpful, called the process's goal and the process's success, that can be inserted into the XPDL file's header. In this way, it is easy to know what processes are similar to each other in terms of the goal or to look only at the successful process and later take into account negative processes. Five labels build the whole index, which is then stored with the XPDL file in the case database. So now we can query the index much easier in the retrieving phase.
Here I proposed a semantical search model because retrieving a BPMN diagram means searching for some similarity related to the content, right? to what is written in the diagram and the diagram's structure. So, I took these two aspects into account. I start with similar semantical cases. The way it works is, I take the whole case database and preprocess the indexes, normalizing and tokenizing the data, and then I feed it into the model. The training phase breaks down the index and learns what kind of phrases and word patterns exist in the database. Then the state of the model is stored. Once a new problem diagram is introduced where the user doesn't know how to fix the problem, it is also indexed and compared to where it would fit in best.
Once the problem is introduced to the model, it will compare the problem diagram's words to the database's preidentified patterns. The idea is that the space of patterns the problem fits in the most can be seen as semantically similar, and the solution in the same area as the problem can be retrieved because they have a similar goal, or they use the same words as they are from the same domain.

Pedro Malta:
When you compare words from one model to the other model, do you have some data that can be the dictionary to guarantee the comparison's correction? Do you understand what I am saying? You have a model with some words, and you want to produce another model with better words, yes?

Philipp Tueschen:
No, basically, I trained the model with all the words and patterns in the database, and I try to find similar cases to the problem. For example, if the goal is to…

Pedro Malta:
…Find what words fit best to the problem.

Philipp Tueschen:
Exactly, I try to group the ones that have similar words, like restaurant, cooking, and frying. When I have a new problem that is about McDonald's, there might be similar cases through the word frying, fitting into the same space of words. Then you can say, okay, they might have similar processes and have some kind of similarity. Those can then be retrieved. This is, of course, all theoretical.

So now we have theoretically semantical similar cases that can be retrieved from the database. In the next step, we can look at the structure of those retrieved cases. Because that is much more computing-intensive, making more sense to have it as the second step. Many articles also show that BMPN diagrams can be easily translated into a graph since it is already basically a graph, right. I propose that we use some form of GED to find similar structures within the BPMN diagrams. Basically, you calculate the minimum edit operations you need to change from the old graph to the current graph. The fewer operations are needed, the lower is the cost and the more similar the graphs are to each other. The more similar the structures, the better it is because the system always has to look for some kind of similarity since the system has to know what it has to retrieve. Otherwise, it is too creative for the system to be able to retrieve something. Based on the graph structures' highest similarity, this one specific BPMN diagram could be retrieved, but there could also be a pool of most similar BPMN diagrams be retrieved and temporarily stored. There might always be something going wrong in the adaptation process, so you have to go through the whole adaptation process again, so it could be faster if you have more similar cases at hand.

Pedro Malta:
Like a validator, yes?

Philipp Tueschen:
Yes, exactly. In the adaptation process, you basically apply the knowledge transfer from the past solution to the current problem. The transformational analogy might be the best way because, for derivational analogy, additional information is needed, which is mostly not available. As before, the GED has already been calculated; at this stage, they only have to be implemented to the current problem. So now we would have a new adapted solution that is successful. But there is a whole larger process for the validation of this new solution. We have a pool of similar solutions from the case database. We adapted the current problem to the old solution, which we now have to store temporarily. Now we haven't used the negative solutions yet. Therefore, similar negative solutions can be retrieved into a separated pool, and in the next step, the temporary solution can be compared to the negative examples. Since they have already some kind of similarity to each other, since they have been retrieved based on the similarity metrics, the idea is that if the new temporary solution is more similar to the negative cases, then before the solution cannot be as successful, so it is rated unsuccessful. So, you have to go back to the pool of temporary solutions and start the adaptation process again. If it is less similar to the negative cases, it must be more successful in one way or the other. The new process can then be validated by translating it back into a BPMN diagram and checking its functionality. If it doesn't function, it is rated unsuccessful restarting the adaptation process. If it is successful, there is still the need to adjust the solution manually where needed. There can be misinterpreted activities, two activities being twice in the diagram, because the old solution's structure has been used on the current problem in the last step without looking at each activity's content. Thus, you have to adjust the content so

that it fits again manually. If this then is found viable, this solution can then be stored with an index where we would go back to the storing phase.

Pedro Malta:
When you speak about negative cases, and you go to the previous slide. When you speak about deleting vertices and edges, do you mean to delete, or do you consider putting the deleted parts in a second class or database that could be used even for the next step of evaluation of the negative part? Do you understand what I am saying?

Philipp Tueschen:
Not really.

Pedro Malta:
What you do here is. You have a problem, you apply these operations, so you will see which of them can be deleted, and you have the new solutions. When in the next slide, you speak about the negative part. Can you go to the next slide? Less similar to negative cases. Could we use the deleted part also in these criteria to validate the BPMN diagram or not? I am saying this. It means consider a process you want to go to a warehouse and buy some products. But you have deleted one of the branches from a warehouse that is not geographical near the organization, so you really deleted it because it is not so useful, more costs, for instance. Could it be used in the case of no stock in the good ones, and you are going to use the other that you already deleted. It could be an interesting question. I should also think about this again. I understand what you a proposing, though.

Philipp Tueschen:
Once it is deleted, it would be gone. You would delete the activity. That is why I said in the end; it is structurally more similar, but not as similar to the negative ones. That is why you validate if you have any problems, so you have to look at it again if it makes sense. It is a very difficult part of finding similar diagrams since you have a structure and what is written in that has a very high value. So, this combination made it difficult to combine. In the end, I didn't get around proposing a manual intervention into the process. But to come back to your point. You mean those parts that have been deleted and then go to the less similar cases…

Pedro Malta:
…Think about it as a recovery plan if you have no options because you chose a good model that doesn't function. There was an explosion, and you don't have any products, so you have to use the deleted part and go to the warehouse that is not near the organization. Even if we have more costs in transport, it could be good. I am saying that when we delete, and we don't use the information anymore, perhaps we don't really have good use of every information that we had in the beginning. Normally, when we draw a BPMN diagram, if you interview 4-5 people, you can draw a BPMN, and I suppose everyone has different visions because everyone

has a personal opinion about the process and the way the process goes. And the consultant, prior to drawing a consensual process a consensual path, is constructed to satisfy the business objectives. So, when you use a tool to separate what you can do and choose the best path, you can have some options about the process; I don't think we should 100% forget what the other visions said. For example, you have a billing process that you want to correct, and there are rules and regulations changed to bill clients. But you have Mister Antonio that works for 40 years in the company, and he knows the history of the process of billing in the company really well, from as it was with the paper to the use of information system now. His definition is not the correct one because the business process changes throughout the different administrations. His role is information, and what he says is linked to the organization's culture because he has so many years of working that we should consider something, even if his path is not the correct one according to the new rules and regulations.
I always think we should consider something from everyone. There are cases we shouldn't consider a person because what he is saying is not useful, but I think we should use something in most cases. Even when you make this validation of this BPMN diagram and even if you delete paths because you have a new solution, the deleted part could be like a backup database, I don't know, that you use in case of some problem within the better solution you used. Do you understand? But I don't want to make your problem more complex. It should be further work for you if you want after the thesis.

Philipp Tueschen:
Would you think that this model, in some sort, would be useful to solve BPMN problems? Once you interviewed 5 or 6 people to structure your process and realize that it doesn't work, you could use some more input, and you know you have 1000 processes that you could use to help you. I mean, this model is not about interviewing people and building solutions from scratch. It was more about the ideas, okay, I have many old solutions, so I could probably use something from the old solution to help me with my current problem.

Pedro Malta:
I think one good point is to retrieve the temporary pool with several examples as the more solutions you have in the database, the more options you have and the better you can adapt the process because you can choose from more cases, so the choice will be more oriented. I suppose the quantity of history should be important to belong to the database.

Philipp Tueschen:
Yes, I feel the adaptation process is still a highly creative process, which is better done by humans than machines.

Pedro Malta:
When we do, for instance, in Covid-19, the problem with vaccine is that we must test what laboratories have in many humans und during many phases so we can be sure that we have a

good vaccine. Here would be the same thing. You can only have few data, and you can try to put better models with a compilation with that data. But if you have more, the probability of choosing the best model is better because you will choose from a larger quantity of solutions, so you must do more correlations and thinking. As a human, we can do this during months and within a computer, perhaps some minutes. If we have a data warehouse a big data warehouse, when we automate when we use technology, it is better for the performance it is quicker, but we must think very good before, because if we don't think very good the computer will do what we say, and the output won't be good.

Philipp Tueschen:
Do you have anything to criticize about the model?

Pedro Malta:
I think this is a good proposal. I would like to see some examples or some data tests.

Philipp Tueschen:
That would be my next step. Assuming I pursue a Ph.D. I would look at the physical implication because right now, this would be too time-intensive to integrate into the thesis.

Pedro Malta:
Anyway, I think it is a good structure here. Theoretical but a good structure. I would advise you to be aware of these visions that we normally forget because we have a better model. I wouldn't forget the other information. I would take care in every step of what we are doing with the validation. So, you can go further with a reason with someone or something verifying what we do. That would be good; for instance, we retrieve a temporary pool of solutions from the case database when you go to the sorted pool of negative solutions. Why are the temporary retrieved temporary pools the bests or good data for this? Can they be sorted as negative or not? There is always a question between the steps to question. And you find in the literature some articles that are part of this model. So, I think you can justify with citation your steps. Even professor Vitor is going to help you. However, if you don't find something, let me know. Maybe I have something that can help you.

Philipp Tueschen:
Thank you, that is very nice of you. Because of this validation step, it was hard to find literature—everything I presented to you as part of my solution, which I build. So, I stopped research at one point. My literature was more based on what is CBR. How can I use GED etc.

Pedro Malta:
Philipp the word validation. Please use it in the methodology part. Validation is more like a reason that you found in articles of other works that justify your steps. There are perhaps some steps that belong to your proposal. So, if it is a new proposal, you won't find anything in

the literature. But some articles could be parallel within this problem of choosing BPMN and trying to better the BPMN designing. I am working in process mining that is a very good exercise to study the process and understand the process and identify as you identify, as you already spoke about, activities and roles and everything. Perhaps some inputs in process mining articles can be useful to justify some steps here, okay. Even in BPM improvement, all the research talks about AS-IS the situation they found in the enterprise, and then let's work to a TO BE that is, for instance, quicker, so let's try to use a simulation too within a BPMN editor and try to reorganize times and tasks and number of requests so we can redesign the process with that focus. So, there are not really validation issues, but they are reasons that justify your model and your steps. Okay. I don't think you need to do this. You already have a good set of research made. So, I don't think you must go to research again for more works, that it. I think you can reorganize. Have you already written anything in your thesis?

Philipp Tueschen:
I am basically all done. Everything I explained to you is all written out. Because it is DSR, I do interviews to validate if my model makes sense. That is the purpose of this interview. So, I would use the outcome of the interview in the validation part of my thesis.

Pedro Malta:
Okay, that was what I was saying. You did your research, and you made this proposal. I suppose during this research, you found some details and reorganize them, and put them as a citation in your text. So, I think you are on a good way. This really makes sense. As you know, research is not always closed. You can always do something more in-depth, so I suppose the important part is to have an idea to test, as you said, in a Ph.D. work. So, it is important to have a conceptual model that can be a solution to a real problem. Okay, so I think you are in a good way. So, I will be available to you if you need anything.

Philipp Tueschen:
Thank you for offering that is really nice.

Pedro Malta:
All right have a good week of work. If you have anything, let me know.

Philipp Tueschen:
I will let you know; thank you. Have a good day.

Pedro Malta:
Bye

Philipp Tueschen:
Bye

INTERVIEW 2 – ISABEL MACHADO ALEXANDRE

Philipp Tueschen:
Thank you. Thank you for joining.

Isabel Alexandre:
Hello, hi.

Philipp Tueschen:
Today is about the final stage of my master thesis. I've used design science research and built a theoretical artifact. Now I'm conducting interviews to validate my model.

Isabel Alexandre:
Okay.

Philipp Tueschen:
My model was about CBR in business process management design. So, my objective was to effectively store, retrieve, and adapt BPMN solutions that fit best the underlying current problem using CBR as a tool. Basically, I built a CBR system that returns past solutions and the depths of these past solutions to a common problem. And therefore, I use the CBR lifecycle as a structure where I had to find a solution to each stage of the CBR lifecycle. And yet what I use then I used to retrieve, reuse, revise, and retain. But I'm starting with storing because when building the whole database. First, I need to know how I can store those BPMN diagrams. So, in the storing step, I first had to translate the BPMN diagram into its underlying XPDL file. Because with this PDF, I can later translate it back into a BPMN diagram for validation. It's then easy to extract information from the XPDL File to build the XPDL file index to query its information later. Going from the diagram to the video file to creating the index, I extract the most important information for the index because it is only effective if it adequately captures and describes the process's content. So here we extract information from the XPDL file for three labels of the index in the context which captures the pool and lane names of the diagram, the keywords which capture the activity names, and the relationships between activities, which are the connections of the BPMN diagrams. So, with these three labels, I grab the entire content of the BPMN diagram. But then there are additional two index labels that give them a tremendous other knowledge about the process. One can add these two labels into this XPDL manually and make one label for either goal. So, it is very interesting to know each process's goal and whether this process is where success or a failure. We can separate those goals later. And those additional two labels, then we would have an index of five labels, which is then stored together with each XPDL file in the overall case database. Now, we have the whole case database with to query for a solution. So, in the retrieving phase of the model,

retrieving analogical BPMN diagrams means we're searching for some degree of similarity related to the diagram's content and the diagram's structure. So here I start with the content of the diagram. And the search model queries the database index for its keywords. And therefore, it uses; first, the whole data in the database normalizes and tokenizes the data. It cleans the index's labels of the database and puts them into the model to train it. And the model is learning all phrases and word patterns that exist within indexes. These patterns are then saved in the mode. Once a new problem is introduced into the model, it will thoroughly compare the words used in the problem diagram to the pre-identified patterns in the database, and the patterns the new diagram fits in the most can then be assumed semantically similar. And with these semantically similar problems, they can be retrieved for the next step, which is the structural similarity. So now we have all the ones where we say okay; they're probably the most similar cases. So now, we need to know whether they're structurally also similar. Therefore, I proposed to use GED to find similar structures. Since BPMN diagrams can easily be translated into a graph already, they are basically a graph. And here, the GED could then calculate the minimum operations needed to adjust one graph to the other graph. And the lower the number of operations, the more similar the graphs are and can be retrieved for the adaptation step. So, the most similar case can then be retrieved. First, we looked at the symmetrical and structural and similarity; we filtered out the semantically similar ones. And now we filter again, this little pool into an even smaller pool, and based on the most structurally similar cases. And then, we use the most similar case to apply the transformational analogy. So basically, we transfer the knowledge from the past solution to the current problem. And as we already calculated the GED, all those changes can then be easily applied from the old solution to the current problem. Although Theoretically, making the current problem a new solution or making the current problem successful and working, there are still many problems when adapting. So, there's a bigger underlying process.

We have all the cases, which are semantically and structurally similar, retrieved in a temporary database. And we adapted from these the most similar solution structurally and saved it in step three, but now, we never checked it against the negative solutions. So, we can sort the negative solutions in an additional pool and compare the new solution to the negative cases.

Isabel Alexandre:
Can you say negative solutions are the ones that are not successful?

Philipp Tueschen:
Yes, unsuccessful solutions.

Isabel Alexandre:
Unsuccessful. Okay.

Philipp Tueschen:
So, in this pool, right, we retrieved all possible solutions. But we never filter out the negative solutions, so now we filter out the negative solutions, and we can compare the negative examples to the current new solution. Now the system is dealing with negative examples; the desired outcome would be a low similarity of the current solution. Because we know negative solutions have some kind of similarity to the current problem, the new solution should be less similar to negative solutions. Because if it would be more similar to the negative cases, the new temporary solution could be seen as unsuccessful as it is more similar to negative cases. And then we would have to go through the whole addiction process again. And that's why I recommended storing a temporary pool so that the adoption process would run a little bit quicker. But once the similarity is less than the negative solutions, the new solution to the current problem must be more successful than before. So in the next step, we would have to validate the solution's functionality, which is why it is transformed back into a BPMN diagram to run if functioning correctly. And if it is successful, one would still need to check manually if activities aren't in the right order, or if the language is according to the current problem, because there can still arise many problems in a creative solution process like this. Once the users found the new solution viable, it could then be retained through the storing face again. Sorry that it took so long, but that's basically my whole thought process of using CBR to retrieve cases, adapt them, and store them again. So yes, that would be my whole model.

Isabel Alexandre:
Okay. Um, you have three different questions. I don't know if you want to ask different things, or do you want me to try to answer such questions? Because I'm not very familiar with the business process management and even notation, so I just picked a little bit. I know what it is, but I don't use it regularly. I don't teach it. So, I'm more familiar with CBR. So, what I may say, maybe not quite a very informed discussion for you, but you are interviewing. So every person from different backgrounds is valuable. Okay, because it's different insights. Right. So, t take into account that I'm not familiar with business process management, I know what it is, but yeah, I have general knowledge. So, we could say, okay, when you say that it might be a good way to solve BPMN problems, what do you mean? It wasn't clear what kind of problems you want to minimize by using things like CBR.

Philipp Tueschen:
When people design business processes, they go to the company, and the company has a problem of wanting to restructure a business process of something. And then, this BPMN specialist is interviewing different people in the departments to get insights, how the process is working, and he is applying it to structure the process, right.
Well, there might be something not working out with structuring the process. So, this database of past problems might have some cases on solving or making it more efficient.

Isabel Alexandre:
Okay. So, you try to provide these kinds of experts with a previous as not previous; we try to identify the domain and possible business process management diagram by giving them this kind of tool. Okay, so my question is, do they use this like, okay, instead of starting from scratch, can I just use these, and it will generate something that might be applied to these, for example meaning the domain, it's the same, and even the activity names might be similar Okay. Is that what you're trying to do?

Philipp Tueschen:
Not from scratch, but one can structure the basic diagram with what you will use certain words; depending on the domain, you will use certain words. So, in medicine, you use different words than in engineering.
The semantic engine will look for similar cases of that domain because it probably will have some similarity. So, it can't build exactly from scratch, but it can analyze a basic diagram and return more complex solutions.
I mean, the specialist probably doesn't know all of the diagrams from the past, but this will help him, like an expert system, for example.

Isabel Alexandre:
How do you plan to evaluate this kind of work? Okay, now you are interviewing about constructing the artifact, all these theoretical approaches you have taken. So how are you going to validate if this is a good solution?

Philipp Tueschen:
Well, basically, the outcome of the thesis would be this theoretical artifact. But the practical way would be to go step by step. So, while researching each phase, storing, retrieving, and adapting, those are very detailed research. So, this is an overall system. But I have to focus first on each sub-section of storing, retrieving, and adapting, and eventually bringing the technical parts together. Okay, because most of the time, when you read about solutions to this problem, it's always about time, time and efficiency. The problem is that the technology's not quite there yet; such as natural language processing for short texts they're very limited. And the adaptation, they use GED actually and look for greedy algorithms, because those diagrams can be very well, they can be very big,

Isabel Alexandre:
yeah, and complex.

Philipp Tueschen:
The mathematical part behind it can get very, very complex. So, it would take a very long time. So, people don't use this much yet. I tried to bring all the ideas together to build an overall system, which practical implications would then be the next pieces.

Isabel Alexandre:
Definitely, okay. Have you tried to apply some examples like having a company or something with some business process management diagram, set a key that you can feed into your solution just to see the outcome of each of these phases that you mentioned?

Philipp Tueschen:
No

Isabel Alexandre:
I think that when you ask me what can be improved, I think that it is always best to find an example and try to use it in your model. I know that you said your goal is to develop this model and to propose it in in your masters but to give more, I wouldn't say value, but it would be much richer if you could find a company and someone that has some of the models and try to use and see how the model would behave when trying to find something similar. It could be like, just one example, but trying to have some cases in your database and giving some examples of how the indexes are created and even the similarity between graphs. Also, saying okay, how do I apply this model to a new company that has this, this, and this is good. And how would it work? Okay, step by step. That's the only thing that I would suggest because, for example, this diagram that you have now on the screen, it's complex to follow, okay? Even phases 1, 2, 3, 4. Now the five can go like bidirectional. So, it's complicated to follow what you say. For you, it's simple, of course, because you have developed it. So, this is my only suggestion to try to get one example. And try to say, for example, this diagram, how it would work, how it would be the transition between those cases. Okay, it can be like simulated cases, but it would give a much clearer idea of all your adaptation steps from the previous solution to the current problem. I think that CBR is a good way to deal with this because it is one of the advantages. But the question is, the semantic that you extract from the diagrams can be limited. And as you said, the graphs can be complex. So sometimes, if you don't have the right cases in your database, it may be difficult to find the most similar one, okay.

Philipp Tueschen:
That's actually one of my limitations because the system always looks for some degree of similarity. There might be cases that have a perfect solution but are not related to the problem at all. So, the system can't find them.

Isabel Alexandre:
Don't even take them. Yeah. Okay. So, these are my only comments. Ah, so I don't know if it was useful, but this is what I see.

Philipp Tueschen:
In general, CBR is about learning from the past. And this is like the theoretical thoughts on how it could be applied practically. Would you say that CBR in this way could be applied to learn from the past?

Isabel Alexandre:
Yeah, I think it is similar. And the question is how you find the similarities between the models. Because as you said, sometimes it is not on the domain, it can have the same structure, but the domain can be quite different. And a human knows, I did this because the business is similar, for example, but I'm dealing with x in this domain, and I was dealing with another thing on the other domain that cannot be compared, but the process and the flow are the same. So, the question here is to find the correct way of finding the similarity between the cases. In the case of life, in terms of architecture or information of information systems, it can be more on the details of the business and the available infrastructure of the kind of activities that the companies do. But it's all about how to compare the cases because we have them stored somehow. So, what we want to do is to find a good way to extract knowledge. And try to because, as you say, I can have a very good solution in one of the cases, but it can be different in terms of domain, and the model will not get it. You could, for example, go first for the graph similarity. If you see the graph's similarity, maybe it can sometimes be, but there you have to, again, to decide if it is similar or not. And then if you apply, okay, now, I have the graph similarity, and I go to the semantic approach, you could just eliminate some of the good solutions. That's the tricky thing.

Philipp Tueschen:
Yes, that was also always my problem. So, I was always looking at what kind of similarities you can use? And my idea was to use first the content and then the structure by tending to find some sort of models that could be applied to find similarities in these areas. So, I always had to break down the problem into smaller problems.

Isabel Alexandre:
Yeah, what did you find in the literature about the problem of finding the similarities between cases? Do they always go in terms of semantics? Or do they prefer the structure?

Philipp Tueschen:
They proposed three areas. It was structure, content, but also behavior. Because it always looked at what happened one step before in the process. And what happens next and after that and so on. So, when I'm cooking, I first have to cook, and then I eat, and this order is very important. I can't first eat and then prepare because nothing will be there. So, they also look at this. But apparently, this was an even more difficult problem to solve. So most papers look at just look at

Isabel Alexandre:
Content extraction.

Philipp Tueschen:
Content extraction, or via this structure itself. And it was always limited to the kind of technologies, and they always focused on very, very specific parts. One was about the structure; one was focusing on the content, but in the end, it was about trying to build the whole system.

Isabel Alexandre:
Also, the watch you do the activity before and after, in your case, this I think it's, it's not impossible, but quite difficult to do. So that would be a difficult way also. Yep, I think you identify the problems and the limitations, and I think that if there is a good solution, it can definitely be found. And if you have practical examples, you could also try to evaluate to know what you have in your database. Like, okay, I'm now trying to find a new solution for something for this problem. Let's see what comes. And with that, you could say, if you had, for example, different domains and cases with different structures, you could try to see the percentage of good solutions found and good solutions not found. I am trying to say if you add a practical and an example or a simulated database, you could do some texts and try to identify the good solution that was not identified by the model. And what were they, the causes, okay, and release, you could like, adding a new step in your model just to try to get them also okay, but that I think it would only come from, like, training your model in a more oriented way. Because without having such cases, you can, for example, imagine that they can exist, but it's difficult to analyze and define. Okay, this is what I have to try to find that solution that is not semantically correct. Not correct, similar, but it's also applied. I think that's because without knowing that it's quite tricky, I think, as it is the combination of semantic similarity and the graphical and graph structure, it's a good answer to your problem. To have a more precise solution, you have to have some cases and analyze what your model would get and what it doesn't get, which could also be useful.

Philipp Tueschen:
I also kept the human factor inside the model, so there will always have to be the person that will have to check.

Isabel Alexandre:
Yeah, yeah, definitely. Or you could have something learning from you, like having a learning model attached to this, but always wait for validation from the human or the expert, whatever you want to call it. Okay. Okay. I don't know if I was a big help. So, I wish you all the luck.

Philipp Tueschen:
Yes, it was a big help with lots of good insights, thank you.

Isabel Alexandre:
I wish you all the luck, okay. It was a pleasure to try to help you. Okay. So, thank you. Bye.

Philipp Tueschen:
Thank you. Goodbye.

INTERVIEW 3 – FREDERICO CRUZ JESUS

Philipp Tueschen:
Hello, thank you for joining.

Frederico Jesus:
Hello Philipp, nice to see you.

Philipp Tueschen:
My topic is CBR in business management design. The objective was to effectively store, retrieve, and adapt BPM solutions that fit the underlying BPMN problem best and adapt it to the current problem using case-based reasoning as a tool. So basically, I was building a case-based reasoning system that returns past solutions and the depths of these past solutions to a common problem. And therefore, I use the case-based reasoning lifecycle as a structure where I had to find a solution to each stage of the CBR lifecycle. And yet what I use then I used to retrieve, reuse, revise, and retain. But I'm starting with storing because when building the whole database. First, I need to know how I can store those BPMN diagrams. So, in the storing step, I first had to translate the BPMN diagram into its underlying XPDL file. Because with this XPDL, I can later translate it back into a BPMN diagram for validation. From the XPDL file, it's easy to extract the information I need to build the index for the XPDL file to query later information. Going from the diagram to the XPDL file to creating the index, I extract the most important information for the index because it is only useful if it correctly captures and describes the process's content. So here we extract information from the XPDL file for three labels of the index in the context which captures the pool and lane names of the diagram, the keywords which capture the activity names, and the relationships between activities, which are the connections of the BPMN diagrams. So with these three labels, I capture the full content of the BPMN diagram. But then there are additional two index labels that give them a tremendous other knowledge about the process. One can add these two labels into this XPDL manually and make one label for either goal. So it is very interesting to know the goal of each process and whether this process is a success or failure. We can separate those goals later. And those additional two labels, then we would have an index of five labels, which is then stored together with each XPDL file in the overall case database.

Frederico Jesus:
Sorry, can I ask a question? Sorry to interrupt, but is that the only information you use from the process model? I think a lot is missing, and you are generalizing. If you are going to get a model from different parts of processes, designing the models is the easiest thing. But on the other hand, I don't think CBR will be helpful. If you could get the distributions, that could be very dangerous. You could be tempted to just okay, this is the same process as some other case in the past, it can be the same process, but the distributions and the parameters of the processes, the arrival of new cases is completely different, which would completely ruin your simulation. You see, it's not like you have some data science software that you think is magic. That's very dangerous because you are applying techniques that you have no idea how they work. And you might as well stay and quit then doing a lousy analysis.

Philipp Tueschen:
Yeah. You're saying that you need all the information to make your proper reasoning for building a whole process; You can't leave information out; you need more.

Frederico Jesus:
I think, listen, I am just talking, you see, I get case-based reasoning for BPM. That is the same thing as consultancy. This happens, but if you focus too much on the models, I would say number one drawing the models is not the real important part of the quantitative analysis. It was the simulation; drawing the models by itself is useful for qualitative analysis. But what you really want to do is a simulation, and you can only run a simulation if you fit the models with all the data you need regarding the activities, the decisions, the arrival rates, the resources, and your salaries. You see. And I think this is very hard to do because of CBR because there each case is literally a case. But at the same time, the ability that you have to export the diagrams if I were you, I would make it clear this is useful for a specific part of BPM, which is qualitative analysis. But as I said, I'm not your supervisor; I'm just here throwing thoughts.
What-if analysis, right? The advantages of using the model from another process are limited because drawing the model is relatively simple. What is more difficult is finding out everything you have to know about each model element.

Philipp Tueschen:
True, the times and the timings of each activity, that they're all run together, later on. That's true. However, when you run into a problem designing the problem, I just focused on how you could use old cases and CBR to learn from old designs. That's all I focused on and used in the retrieving phase. I used semantic analysis and GED to analyze the structures and the diagrams' content to retrieve a correct graph.

Frederico Jesus:
If you could add to that the AS-IS and the TO BE, you are leaving the simulation results because listen, if you think of a BPM business consultancy, what you want is not retrieved. You want to retrieve the work and the thoughts of the process analysis to improve the business process. I think they're to add more value. But listen again, I am just throwing out some thoughts. Listen, I have a retail company in Portugal, for example. And I have a problem because of the delivery's digitization. Well, but there was another retailer that redesigns the delivery process by incorporating technology, so I don't need to reinvent the wheel. What do I want from that? I want the AS-IS and eventually the report with suspending the analysis. That's what I want.

Philipp Tueschen:
So basically, I store with the old TO BE solutions correct. So, I store the final solutions of the process. So, how in the past they made the problem work, the problem process work. And I have successful and unsuccessful cases in the database.

Frederico Jesus:
But the AS-IS isn't too important, right? Because you can have two AS IS, and each one leads to one TO BE, and what eventually is your choice for the most appropriate TO BE will be the AS-IS that is more similar to what you have. But I interrupted your presentation. I'm sorry.

Philipp Tueschen:
Oh, no worries. So, there are two AS IS, and they go into one TO BE right.

Frederico Jesus:
No, what I mean is this. Imagine you have company A that has an AS IS and a TO BE? You have a company B with an AS IS and a TO BE, and you are at company C. So, you are interested in getting their TO BE's of A and B. That's the most important, but it can also be essential to see which company was originally more similar to what you have now. Do you know what I mean?

Philipp Tueschen:
Okay, so you say that you look at C, and you check which one from A and B are more similar to C?

Frederico Jesus:
Yes. In the beginning, that might give you some hints as to what is the better TO BE.

Philipp Tueschen:
Yes. That's actually what I also later then do as well. So, in the retrieving phase, I looked at it more from the perspective, now I stored all the cases from BPMN as XPDL and have all the indexes, I train a text-mining tool looking at a semantic space of all the indexes. So, you just look at the content, and you want the more similar cases. So, you look at the model that is

then finding out, under unsupervised learning, some word patterns and phrases used around the different labels. And then, when you insert a new problem, you also build the index, and you look where the new problem fits in the most. And then, you fit the new problem in the semantic space with the patterns. And the cases around the new problem could then be seen as semantically similar because they use similar phrases, and probably in the domain like a hospital, you use similar words. So, it is in the same area. And this one you would then retrieve from the case database. And then, because you narrow down the similar cases you have, you could use a GED to find within similar cases similar structures. Here we also have to keep in mind that those structures can be positive or negative. So, I retrieve positive and negative cases.
I store multiple of these in a pool. So now I have diagrams that are theoretically similar to my current problem altogether in one pool. And I use the most similar problem, apply the graph at a distance. And now I would have a new adapted solution theoretically, according to Carbonell and his theory of transformation analogy. And this new solution should work. But because you have changed the old graph, there might be problems, right? You don't know about the content of the whole new graph or the entire new diagram. So, I looked at it as some kind of validation process, so I have the pool of temporary solutions, then I adapt the best one to the current problem in the adoption process, and then I store this new solution. After, I compare the new solution to the negative solutions I retrieved. And my assumption was when the new solution is more similar to the negative cases, that would indicate a negative or an unsuccessful solution. But if it would be less similar to the negative cases, that would indicate it should work. But you don't know yet if it works. So, you would have to translate it back into the BPMN diagram and check for its validation if it's running or not. And that's where then the manual or the human intervention comes in, where a person has to check whether the new solution works or not. If it's not working, you would go back to the pool and retrieve another solution. So, this is some sort of support for designing a new BPMN diagram because, while you design a process, there are still 1000 10,000 processes somewhere that could give you some help in creating it more effectively, for example. And then, once the solution would be, according to the needs, you could retain it again, as in the store and process.

Frederico Jesus:
And you are looking to implement this?

Philipp Tueschen:
Well, that would be the next step. Yes. But the idea of the thesis was to use CBR and BPM. And these were my thoughts on how you could make it work. Because it was about retrieving old cases to learn from, it was about learning from the past rather than building a new process. You usually interview at the company, like how does the process work, right. You build it according to what the company says, and then you go from there.

Frederico Jesus:
It seems very nice. You have a sophisticated part that has to do with the research and how you search for previous models, which is more an area of data science, right? More than BPMN. I think this is interesting. But this is then via underlying rudimental companies, right. The same method, the same process is already then in knowledge management, am I wrong? For example, in consultancy companies with knowledge management, the idea is trying to get the same objectives as you propose here, which is not reinventing the wheel in one sentence, right. But it's done in a more in an old-fashioned repository. Here, what you're saying is okay, so every consultancy or BPM project aims to improve the way companies do the business that is their work, right. Because we have BPM and models, we can enhance the knowledge management part by using the BPM and models to search for similar projects in the past, right.

Philipp Tueschen:
Exactly.

Frederico Jesus:
I think it's a very good idea. My only concern, I guess it's congratulations. I think looking through BPM lenses, and there are two things to keep in mind: the models are much more than the elements. The BPMN models are a means to an end; they are not the means by themselves. A model is much more than the elements. So also, the metadata that comes with them, for each activity, right. And that is the most difficult part to get for CBR, and the most difficult part, from what I understand, to put in that file system you are using.

Philipp Tueschen:
Yes, exactly. So, I had to eliminate this kind of information to be able to compare the cases.

Frederico Jesus:
Yeah. And it makes sense because a clerk in Portugal is not the same salary as a clerk in Germany. If you add that kind of metadata in your files, people could make mistakes because they will not even look at it. So, what I eventually suggest is that along with the model, when someone finds a similar project in the past, you say that everything else goes along with that project. The reports, the qualitative analysis, the quantitative analyses, and the case description because you need a picture of the best project.

Philipp Tueschen:
That's an excellent idea. So, without saying, okay, you analyze all this data to make it even more accurate to retrieve it more precisely; you retrieve that additional information to have more information for a person to interpret.

Frederico Jesus:
Yes. My point is this, Philipp. The models by themselves are not useful enough. If you give me a process model often accompany similar to the one, I'm doing BPM project, the models will be almost useless. Even the TO BE because for me to see what is the TO BE, I need to see what the original starting point was. And I need to have the justification for what changed in that case because not everything will adapt to my ongoing project. Think about the laws, regulatory pressures, and contexts that companies need to follow in the USA but are not applied in Portugal. You see. If you don't have all of this with you, and you focus excessively on the model, I think you might be missing a big point.

Philipp Tueschen:
Right. That's a good point. I probably missed that. I focused more on the CBR site than on the BPM site. It was more like how and what can I apply it most.

Frederico Jesus:
Yes. If I give you an AS-IS process, just the model without all the other metadata, no, but if I just give you the AS-IS and TO-BE model, maybe. I don't know if you only consider this, but this is not very useful. So I will get it and say I don't have any idea what was done here.
And if I don't have any idea, what is it of use for me and the project I might be working on?

Philipp Tueschen:
Right. Yeah, I'll consider that. That's probably a good part to add in the main part.

Frederico Jesus:
But I think it is very interesting. You perhaps can turn it into software and sell it—a Knowledge Management System or something.

Philipp Tueschen:
Well, that's where it comes from, right. Knowledge Management, expert systems, all this. So, you would say that the process like this could be working actually, from a knowledge perspective?

Frederico Jesus:
Yeah. As I said, I think this kind of system, but manually. They are used more by international companies. Okay, but perhaps you can talk with someone that works in consultancy, but I am pretty much sure that in consultancy, you do this. You will not, every time you have a project start from scratch.

Philipp Tueschen:
So it's more from a knowledge perspective, less from a BPM perspective.

Frederico Jesus:
I am not sure you sell it as Knowledge Management; I think BPM is sexier and more appealing. Yes, I'm trying to say that a similar system or approach is used in knowledge management. I believe that the terms you used are good because it's BPM, right? Because you're using process models. I'm just not an expert on this, so I'm just giving some rudimentary thoughts.

Philipp Tueschen:
They were very insightful. I am almost done with the whole thesis, and it is all written out. Now it is the validation of the model, whether it works or not.

Frederico Jesus:
But listen, don't let my comments make you change what you have. Especially if you're already finished, I'm just giving some insights and listen; you know it much better. I am only here for 10 minutes, and then I say things that you can feel free to ignore and say this does not make any sense. You see, so if you have things and I'm sure you already have. So, don't let anything that I say make you question your work. Eventually, just some of these comments can work as makeup, helping to wrap up for future research.

Philipp Tueschen:
Thanks a lot for your help. Thank you. Have a great holiday season, although, with Corona, it won't be as lovely.

Frederico Jesus:
Thank you. Bye.

Philipp Tueschen:
Bye.